HAL
open science

Analysis and Simulation of Optimal Motions in Rock Climbing

Simon Courtemanche

To cite this version:

Simon Courtemanche. Analysis and Simulation of Optimal Motions in Rock Climbing. Graphics [cs.GR]. Université de Grenoble, 2014. English. NNT : 2014GRENM082 . tel-01395561

HAL Id: tel-01395561

https://theses.hal.science/tel-01395561v1

Submitted on 10 Nov 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Simon COURTEMANCHE

Thèse dirigée par **Lionel REVÉRET**

préparée au sein **du Laboratoire Jean Kuntzmann (LJK)**
et de **l'école doclorale EDMSTII**

Analyse et Simulation des Mouvements Optimaux en Escalade

Thèse soutenue publiquement le **20 octobre 2014**,
devant le jury composé de :

M. James L. CROWLEY
Professeur, Grenoble INP, Président
M. Franck MULTON
Professeur, Université de Rennes 2, Rapporteur
M. Laurent GRISONI
Professeur, Université de Lille 1, Rapporteur
M. Franck QUAINE
Maître de Conférences, Université Joseph Fourier, Examinateur
M. Thomas ROBERT
Chargé de Recherche, IFSTTAR, Examinateur
M. Lionel REVÉRET
Chargé de Recherche, Inria, Directeur de thèse

Résumé

À quel point les mouvements humains sont-ils optimaux ? Cette thèse aborde cette question en se concentrant particulièrement sur les mouvements en escalade, étudiés ici sous trois aspects complémentaires que sont la collecte expérimentale de séquences de grimpe, l'analyse biomécanique de ces données, et la synthèse de gestes par optimisation temporelle. La marche fut l'objet de nombreux travaux, avec de bons résultats notamment en animation [Mordatch 2013]. Nous nous intéressons ici spécialement au problème original des mouvements d'escalade, dont la diversité et leur caractère multicontact présentent une complexité intéressante pour l'évaluation des caractéristiques du mouvement humain. L'hétérogénéité du répertoire gestuel rencontrée en escalade s'explique par plusieurs facteurs que sont l'évolution sur des parois de formes variées, la multiplicité des niveaux d'expertise des pratiquants, et des disciplines différentes au sein même de l'activité, à savoir le bloc, la difficulté, ou encore l'escalade de vitesse. Notre démarche d'exploration de ce sport se décompose en trois étapes : la collecte de données par une capture de mouvements multicaméra avec marqueurs, couplée à un ensemble de capteurs de force montés sur un mur de bloc en laboratoire ; une analyse du geste par dynamique inverse, prenant exclusivement des données cinématiques pour entrées, basée sur une minimisation des couples internes pour résoudre l'ambiguïté du multicontact, intrinsèque à l'activité d'escalade, validée par comparaison avec les mesures capteurs ; et enfin, l'utilisation d'un critère d'efficacité énergétique pour synthétiser la meilleure temporisation associée à une séquence de déplacements donnés. Les enregistrements expérimentaux se sont fait à l'université McGill qui dispose d'un mur instrumenté de 6 capteurs de forces, et d'un dispositif de capture de mouvements 24 caméras, nous ayant permis de collecter des données sur une population de 9 sujets. L'analyse de ces données constitue la deuxième partie de cette thèse. Le défi abordé est de retrouver les forces externes et les efforts internes à partir uniquement des déplacements du grimpeur. Nous supposons pour cela une répartition optimale des efforts internes. Après analyse, cette répartition s'avère être plutôt uniforme que proportionnelle aux capacités musculaires des différentes articulations du corps. Finalement, dans une troisième et dernière partie, nous nous intéressons à la temporisation des gestes en escalade, en prenant en entrée la trajectoire du grimpeur, éventuellement issue de cinématique inverse pour s'affranchir de la nécessité d'une capture par marqueurs et caméras infra-rouges. En sortie, une temporisation idéale pour cette trajectoire est trouvée. Cette temporisation s'avère réaliste, mais manque d'une modélisation des instants d'hésitation et de prise de décision, ainsi que d'un modèle d'établissements de contact, phénomène présentant un délai temporel non pris en compte pour l'instant.

Abstract

How optimal are human movements ? This thesis tackles this issue by focusing especially on climbing movements, studied here under three complementary aspects which are the experimental gathering of climbing sequences, the biomechanical analysis of these data, and the synthesis of gestures by timing optimization. Walking has been largely studied, with good results in animation [Mordatch 2013]. We are interested here especially in the original question of climbing motions, whose diversity and multicontact aspect present an interesting complexity for the evaluation of the human motion characteristics. The heterogeneity of climbing gestures can be linked to several factors which are the variety of wall shapes, the multiplicity of climber skill levels, and different climbing categories, namely bouldering, route climbing or speed climbing. Our exploratory approach of this sport consists in three steps: the data collection by multicamera marker-based motion capture, combined with a set of force sensors mounted on an in-laboratory bouldering wall; a gesture analysis by inverse dynamics, taking only kinematic data as inputs, based on the minimization of internal torques to resolve the multicontact ambiguity, intrinsic to the climbing activity, validated by comparison with sensor measurements; and finally, the use of the energy efficiency criterion for synthesizing the best timing associated with a given sequence of movements. Experimental recordings were made at McGill University which has a climbing wall instrumented of 6 force sensors, and a motion capture device of 24 cameras, which allowed us to collect data on a population of nine subjects. The analysis of these data is the second part of this thesis. The addressed challenge is to find the external forces and internal torques from the climber's movements only. To this end we assume an optimal distribution of internal torques. After analysis, the distribution turns out to be rather uniform than proportional to the muscle capacity associated to each body joint. Finally, in a third and last part, we focus on the timing of climbing gestures, taking as input the path of the climber, possibly after inverse kinematics in order to overcome the need for a capture with markers and infrared cameras. As output, an optimal timing for this path is found. This timing is realistic, but lacks of a modelization for hesitation and decision making instants, as well as a model for the contact establishment, with the associated temporal delay currently not taken into account.

Contents

1 Introduction **1**
1.1 Context and Motivation . . . 1
1.1.1 What is climbing . . . 1
1.1.2 Benefits and needs . . . 3
1.1.3 Computer science and climbing . . . 3
1.2 Positioning and targeted issues . . . 4
1.2.1 Physical simulation of human motion . . . 4
1.2.2 Challenge for a biomechanical model . . . 5
1.2.3 Physically-based motion analysis from video . . . 6
1.3 Addressed Climbing Specificities . . . 6
1.3.1 Holds and their quality . . . 6
1.3.2 The observed motion factors . . . 8
1.3.3 Energetics . . . 10

2 Climbing Data Capture **13**
2.1 Related Work . . . 13
2.2 Bouldering Data Capture . . . 16
2.2.1 Hardware and softwares, synchronization . . . 16
2.2.2 Frames, wrenches, and their manipulation . . . 25
2.2.3 Sensor frame calibration . . . 32
2.2.4 Capture sessions . . . 37
2.3 Top-Rope 2D-Video Capture . . . 42
2.3.1 Capture protocol . . . 42
2.3.2 3D reconstruction of wall . . . 42

3 Inverse Dynamics with Contact Force Prediction **49**
3.1 Introduction . . . 49
3.2 Related Work . . . 51
3.3 Inverse Dynamics Formulation . . . 52
3.3.1 Equation of motion . . . 53
3.3.2 Disambiguation by minimization of torques and forces . . . 54
3.3.3 Contact transition model . . . 56
3.3.4 Joint torque weighting . . . 58
3.4 Experiments and Validation . . . 58
3.4.1 Standard inverse dynamics examples . . . 60
3.4.2 Climbing with limb specific weights . . . 61
3.4.3 Climbing with anatomical weights . . . 62
3.5 Conclusion . . . 71

4 Motion Synthesis by Timing Optimization **73**
4.1 Introduction . . . 73
4.2 Related Work . . . 75
4.3 Problem Formulation . . . 77
4.3.1 Continuous timing optimization formulation . . . 77
4.3.2 Trajectory sampling to define discrete optimization variables 79
4.3.3 Timing resampling for inverse dynamics . . . 80
4.3.4 Timing cost . . . 82
4.4 Results . . . 87
4.4.1 Bouldering motion . . . 87
4.4.2 Input parameters sensitivity and setting . . . 96
4.4.3 Long sequence . . . 117
4.5 Conclusion . . . 140

5 Discussion and Perspectives **145**

Bibliography **149**

Chapter 1

Introduction

Contents

1.1 Context and Motivation **1**
1.1.1 What is climbing 1
1.1.2 Benefits and needs 3
1.1.3 Computer science and climbing 3
1.2 Positioning and targeted issues **4**
1.2.1 Physical simulation of human motion 4
1.2.2 Challenge for a biomechanical model 5
1.2.3 Physically-based motion analysis from video 6
1.3 Addressed Climbing Specificities **6**
1.3.1 Holds and their quality 6
1.3.2 The observed motion factors 8
1.3.3 Energetics 10

1.1 Context and Motivation

1.1.1 What is climbing

Climbing consists in moving on a given wall. The wall contains various protrusions called the holds. The given path to follow is called the route. Each route has its own difficulty which depends on the gestures involved by the holds, which can be more or less energetic or complex, according to the slope of the wall, the size and the shape of the holds, their spacing, and their respective positioning. Each climber has his own muscular limits, gesture knowledge and preferences, making the perception of the difficulty of a route relative to each climber. Nevertheless, a common scale of difficulty for climbing routes can be established by averaging the individual perception of a population of climbers. For a given route, for a given climber, the question of finding the sequence of movements, or moves, that will enable the climber to reach the last hold is called a climbing problem. Each climbing problem has several solutions that are more or less athletic, or require more or less suppleness. If the climber is climbing at his maximum level, then he has only one solution for sending[1] the route. An example of such a situation is

[1] sending: solving a climbing problem

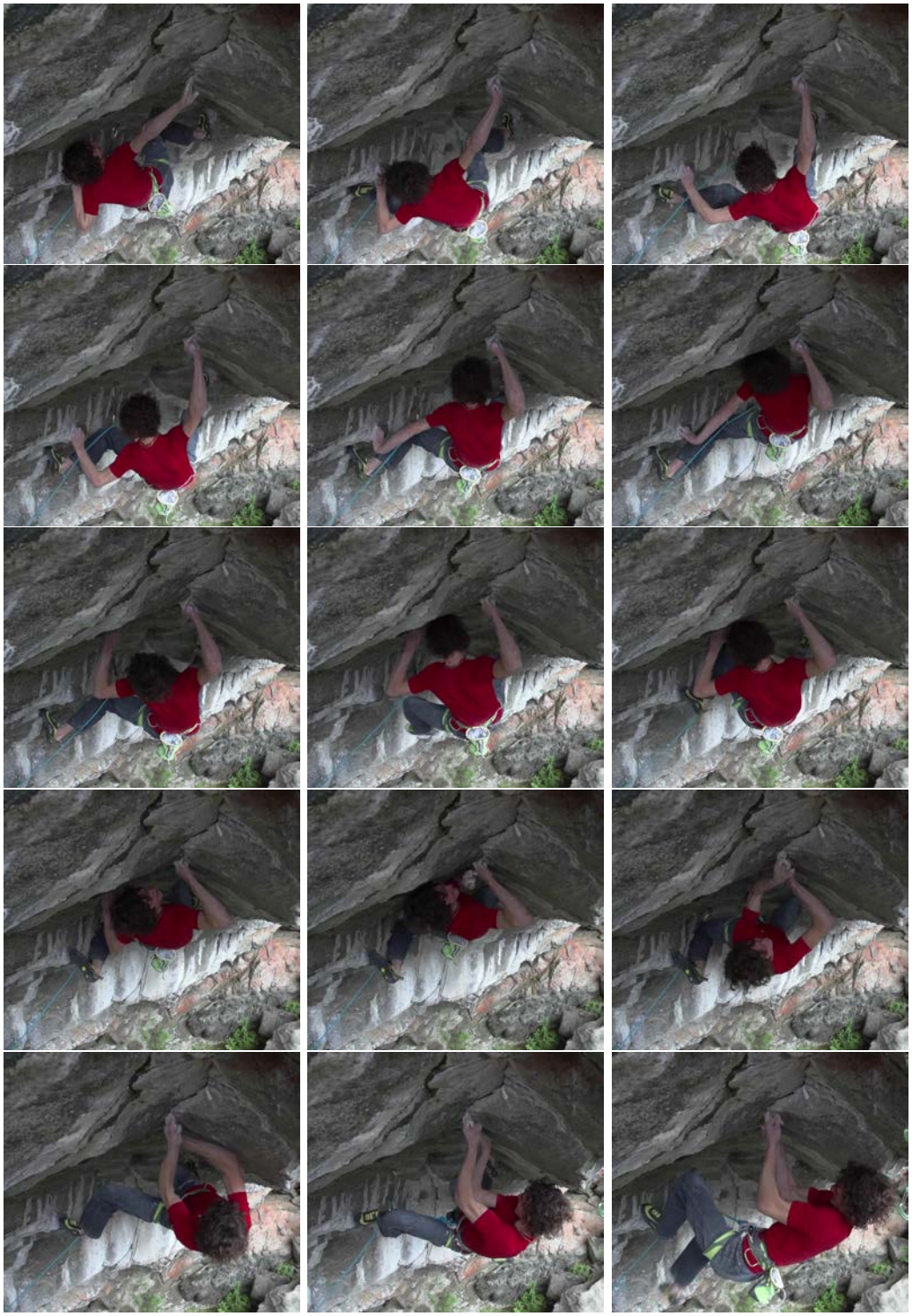


Figure 1.1: *Adam Ondra, world best climber, in the crux of Change, 9b+, world hardest route. The only sequence of moves that works for this climber is ruled by the environment shape and the laws of physics. Images are used with the kind permission of their author Petr Pavlíček.*

shown figure 1.1. One way to reach this solution is to tune the sequence of moves to make them as efficient as possible for the given problem, in order to have enough energy for the whole ascent.

1.1.2 Benefits and needs

Climbing is a sport and has therefore all the benefits mentioned for instance by the Secretary-General of the United Nations (U.N.), Ban Ki-moon: "Sport is increasingly recognized as an important tool in helping the United Nations achieve its objectives, in particular the Millennium Development Goals. By including sport in development and peace programmes in a more systematic way, the United Nations can make full use of this cost-efficient tool to help us create a better world." Via the U.N. Office on Sport for Development and Peace (UNOSDP), the United Nations promote sport as a powerful vehicle for positive social change and a means to promote education, health, development and peace. According to the Sport for Development and Peace International Working Group (SDP IWG), sport is seen to have the most benefits in:

- Individual development
- Health promotion and disease prevention
- Promotion of gender equality
- Social integration and the development of social capital
- Peace building and conflict prevention/resolution
- Post-disaster/trauma relief and normalisation of life
- Economic development
- Communication and social mobilisation

As a sport, climbing offers the advantage of maintaining the physical and mental well-being. It also enhances the personality of an individual, is a good source of entertainment and offers health benefits such as lowering the risks of diabetes and heart diseases.

There are vertical locomotion needs present within the professional markets. Example areas include rope access and confined spaces, tree care, energy and network, framing and roofing, and vertical rescue. Training is needed in these areas, and this thesis is also aimed to provide a visual support for professional training to understand the forces involved while operating in various vertical environments.

1.1.3 Computer science and climbing

By including a biomechanical analysis, a climbing simulation environment could be used for professional training in work-at-height and rescue activities, especially to

prevent work injuries related to positioning. Indeed there exist currently no analysis nor visualization tools to find and explain the best positioning for moving vertically. Visualizing efforts needed to perform a given motion would help to explain difficulties encountered when climbing, and would improve the understanding of this sport. Such a tool could be used as teaching support, and could help in the prevention of injuries, which occur too frequently in this sport.

A climbing video game could be used to prepare climbing sessions. To that end, a robust physical analysis of climbing motion is needed, as the one proposed in this thesis. A focus is made on the gesture optimality, such that performance improvements can be targeted. The optimality aspect of sport climbing is usually tackled by athletes on a feeling level, and only a few books in sport science explain climbing gestures on an efficiency point of view [Hague 2006]. The optimal gesture issue is currently receiving high interest in both computer graphics and robotics. This thesis takes advantage of this ideal opportunity to bring computer science technologies into sport science in order to bring a computational support for performance improvements.

The motion generation issue is shared by climbers and by graphics artists. In both of these domains, the question of finding the human movement sequence in response to a given environment and task can be met. We propose in this thesis a solution based on energy minimization in the specific case of a vertical environment, which provide a way to automate motion generation. In the continuity of this thesis, the converse problem could also be tackled: given an available energy level, what environment can be climbed. Answering this question of computer assistance in the design of climbing routes would help to adapt them to a wider public, such as children and people with physical disabilities, for whom routes are not easy to design.

1.2 Positioning and targeted issues

1.2.1 Physical simulation of human motion

Physically based character animation is an active domain in computer graphics. The challenge is to produce natural motions of different activities that have the attributes and subtleties of real human motion. Hence, data driven methods are popular approaches to generate natural motions, and feedback controllers have been designed for successful balance during stepping [Yin 2007] and standing motions [Macchietto 2009]. In the absence of motion capture, characters can still be animated by other means, and some have proposed to control animation using low frequency vibration modes of a character's mechanical model [Kry 2009]. Optimization is another method widely used for generating plausible motion of animated characters by minimizing energy and other objectives [Nunes 2012]. Dealing with complex contact changes during optimization is a major challenge, which can be addressed by breaking a motion into phases with simpler goals [Al Borno 2012], and with the use of continuous auxiliary variables to help convergence [Mordatch 2012]. Optimization-

based methods aim to produce plausible motion while focusing on only a small set of important features [de Lasa 2010]. With minimal tuning, dexterous manipulation and grasping can be animated from a single input pose [Liu 2009]. Constrained optimization has also been used to design controllers through higher-level objectives, for instance, contact constraints to generate a climbing controller [Jain 2009]. Mid-level control structures are also useful in repetitive motions, such as certain grasping and manipulation tasks [Andrews 2013]. Motion capture and force sensing was used for capturing and resynthesizing grasping motions [Kry 2006]. [Aladdin 2012] started the work on data driven climbing by focusing on optimization of static poses using captured force and motion, and we are extending this work for dynamic motions. Only few static poses can be simulated. In this thesis, entire climbing motions are generated smoothly.

1.2.2 Challenge for a biomechanical model

In standing bipeds, as in quadrupeds, the displacement of a body segment is accompanied by muscular activity and mechanical changes involving other segments, which contribute to the control of body balance. In rock climbing, the understanding of how these forces are shared among the holds requires the solution of an under constrained problem, and this represents a real challenge in biomechanics [Delp 2007]. Previous results obtained with restricted simulated rock climbing movements have shown that the supporting forces are controlled differently in the vertical and horizontal directions [Quaine 1999]. During a single limb release without any dynamic requirements, both vertical and horizontal forces increase on the side contralateral to the moving limb, whereas only horizontal forces decreased on the ipsilateral hold. This specific force sharing has been presented as the most appropriate motor strategy to ensure body balance with minimum energy expenditure [Noé 2001]. One aspect of this thesis is to extend these studies to more realistic simulations of the whole body during the action of vertical locomotion. This has required the development of new biomechanical approaches that combine whole body modeling and optimization techniques to solve the support redundancy.

The numerous finger degrees of freedom and hand muscles allow the hand to adapt to the hold shape, and thus to perform a large number of gripping techniques [Schweizer 2001]. Hence, depending on how handles are manipulated, supporting forces are transmitted differently in the body. The capacity to grasp holds is the prime factor for increasing the quality of the whole body balance, and the efficiency and safety during rock climbing. To date, hand biomechanical analysis in rock climbing has focused only on injuries with simulated hold requirements or on cadaverous hands [Schöffl 2007]. No studies have been conducted during an actual rock climbing movement. From laboratory tasks with the hand flat on a table and fingers placed in adapted force devices, it is well documented that the forces exerted by the fingers are not different between the “slope” and the “crimp” grip techniques, whereas the addition of the thumb in the “crimp” grip allows for an increase of 20% in the supporting force [Quaine 2011]. However, nothing is known about the

hand biomechanical behaviour during actual climbing. The large variation of size and accuracy, from hand to whole body, is an important scientific and experimental challenge.

1.2.3 Physically-based motion analysis from video

From monocular view to multiple camera systems, numerous works have dealt with human motion tracking. Moeslund and Granum have provided a good survey on this topic [Moeslund 2001]. We recently proposed an innovative method using multiple cameras [Duveau 2012]. Our approach is based on the learning of a set of poses. It has shown to reach state of the art accuracy on the commonly accepted evaluation database HumaneEva in the computer vision community. Being based on the learning of a set of poses, it will be necessary to adapt to an optimal set of poses for climbing. The other set of related works is the usage of physics-based modeling for information retrieval from video. Salzmann and Urtasun showed the benefit of using physically-based model for the video tracking of simple objects [Salzmann 2011]. However, their framework is not adapted for our goal as we require the extraction of complex motion of the whole body as well as contact forces. Brubaker et al. presented a method to extract parameters of a physical model from optical data [Brubaker 2009]. The drawback of this method is that it uses a penalty-based contact method which is an oversimplification of the contact physics. In this thesis, we used a constrained-based method, and its accuracy has been proven by comparison of the estimated contact forces with contacts measured at the climbing wall from the animation lab in McGill University, Montreal. Similarly to [Vondrak 2012a], we would need to couple these results using a physics-based simulation with our optical motion tracking works.

1.3 Addressed Climbing Specificities

1.3.1 Holds and their quality

Solving a climbing problem means finding the sequence of movements to climb a given set of holds. The shape of the holds influences drastically the motion performed by the climber to solve the proposed climbing problem. The figure 1.2 illustrates this concept. The difference between a good and a bad hold lies in the number of muscles needed to take this hold, along with the amount of strength these muscles must generate. For instance, if the hold is good enough to allow the hand palm to be in contact with the hold and thus to generate more friction forces, the muscles activating the fingers tips will be relieved, and conversely, if only the finger tips are in contact, the climber will make stronger demands on the finger muscles to use this contact. Therefore, the climber avoids using holds that require more grip strength than others, and consequently the observed body motions are influenced by the type of hold available for the climb.

A similar phenomenon can be obtained when trying to walk or to run on toe

(a) *jug* **(b)** *micro edge* **(c)** *3-finger pocket*

Figure 1.2: *Three different types of hold. The jug (a) is the easiest hold to take. It allows the climber to use the friction of the hand palm and the friction of all finger phalanges, therefore reducing the effort needed by the forearm muscles to activate the finger joints. On the contrary the micro edge (b) requires the transmission of the contact forces through all the finger joints. On this picture we can also see the clear delimitation of the white area that appears after the drying of a liquid chalk application, used to dry the finger perspiration. On the background, the magenta crash pad, put on the characteristic Fontainebleau sand, is used to absorb the possible fall of the climber. Finally, the orientation of the hold can also affects the difficulty of the grapsing. For instance a 3-finger pocket is more difficult to grasp in a roof (c).*

tips, without heel contact. Much more energy is needed to transfer the whole body weight through the toes until their tips if the heels do not touch the ground, than if this transfer is stopped at the heels when they are in contact with the ground, involving that the toes bear almost no weight. If the walker or the runner had to go as far as possible, and had the choice, he or she would certainly choose to use his or her heels, and this choice would affect the observed motion, similarly to the previous explanation.

To sum up, to be able to climb as high as possible, climbers try to avoid using holds requiring a powerful grip. In other words, climbing motions are adapted to the shape of the holds available for the ascent.

1.3.2 The observed motion factors

Climbing motions are the result of several factors that are specific to this activity. To understand these factors, we will explain in more details the process of climbing. We then list the motion factors we have identified. In this subsection we explain the choices we have made for recording in and out-of-laboratory climbing motions, on a bouldering wall and on a free climbing route.

Climbing disciplines We distinguish three main climbing disciplines: bouldering, top-rope climbing and lead climbing. The first one consists in climbing with no harness a wall about three meters high. The two others correspond to climbing with a rope, and are gathered under the term free climbing. Free climbing is the opposite of aid climbing. Aid climbing is a special climbing discipline older than the free climbing where any artificial aid, such as ladders or hooks, can be used to progress on the wall. All these disciplines do not take into account the time needed to climb the route or the boulder, but only the difficulty of the climbing problem. On the contrary, speed climbing is the discipline focussing on the ascent time. In the following, we define in more details the terms top-rope climbing and lead climbing, and explain why we have chosen to collect top-rope climbing data instead of lead climbing data.

Top-rope climbing Top-rope climbing is a way to climb routes taller than bouldering problems, which makes the fall dangerous without safety equipments. The process is to pass the rope through a karabiner fixed at the top of the wall. The climber ropes up at one extremity of the rope, and the belayer passes the other extremity of the rope through a belay device. The rope is maintained tight enough by the belayer, such that any hypothetical climber's fall can be safely stopped before reaching the ground.

Lead climbing Lead climbing is similar to top-rope climbing, with the key difference that the rope is not passed through a karabiner at the route top, but must be clipped to the wall by quickdraws every two or three meters during the climber's ascent. The figure 1.3 illustrates this clipping process. If the climber releases the

Figure 1.3: *The clipping process, occurring while lead climbing, disrupts the continuity of climbing movements.*

holds, then a free fall phase starts, stopped by the belayer when the rope finally tightens, thanks to the last quickdraw that the lead climber has set. This potential free fall phase makes the lead climbing more difficult to study than top-rope climbing, because having in mind this possibility, the climber can make safer movements than he or she would ideally do to climb the route. This psychological stress and anxiety is mentioned in [Sheel 2004] to be an important component of rock climbing. The second disruptive factor of lead climbing is the quickdraw settings, which makes the climber switching from the climbing task to a securing task. To remove these two factors that could disrupt the observed motion, we choose to study top-rope climbing.

Chalk usage Another factor that influences climbing motions is the use of powder chalk to remove hand perspiration. Indeed, during the climbing effort, hands have a tendency to perspire, which reduces the friction coefficient between the hands and the holds, making the climb harder. This issue is often overcome by bringing a chalk bag for the climb, attached on the back of the harness. When hands perspire, the climber moves them one after another inside this bag to dry them with the chalk. In order to avoid those off track motions, we asked the climber not to use chalk. Note that liquid chalk could also have been used here, but the holds have been chosen easy enough to avoid chalk at all.

On-sight and red-point The next identified factor is the knowledge of the route before the climb attempt. When the climber makes his first attempt in the ascent of a given route, we say that he is climbing on-sight. If the climber does not success to climb the route on-sight, he or she usually tries to improve the movements in the following attempts in order to make them more efficient, to be able to climb the route more easily. Finally, when the climber successfully climb the route after several attempts, the result is call a red-point. Redpoints have been brought in the mid-1970s by Kurt Albert, a mathematics and physics teacher from the Frankenjura, who painted red points at the bottom of the aid climbing routes he managed to free climb. Usually on-sight and redpoint attempts implicitly assume that the climber is on lead. Because similar phenomenons can be observed with the first and the

next top-rope attempts, we will also use the terms on-sight and redpoint for top-rope climbing. The on-sight motion factor has unfortunately been identified after the establishment of the capture protocol. Therefore, we mainly observed on-sight climbing runs, which contains hesitation phases and motions that could have been more efficient after several tries.

Cameras The last factor we identified is the presence of cameras, which can increase the stress of the climber. Under the cameras, the observed motions are not as natural as motions of free training sessions. A common solution used to that problem in sport training is to bring the camera at each training sessions such that the climber can be used to it. But this solution would have required more time than the time we had for the recordings.

1.3.3 Energetics

In this subsection, the energetic model used in sport science for high-level climber training is compared with the model used in recent animation works, which has been borrowed from the biomechanics literature, and with the model used in this thesis.

Energetics in sport science Three exercise energy systems exist in climbing [Guyon 2004]. The first one, the alactic anaerobic energy system, is involved in short and intense efforts, typically less than $10s$. The second one, the lactic anaerobic energy system, is involved in medium efforts, with a duration between $15s$ and $3min$. The last one, the aerobic energy system, is involved in lower intensity effort which can last up to several hours. These three energy systems operate simultaneously, in different proportions according to the nature of the climb, and according to the level of the climber. [de Moraes Bertuzzi 2007] showed that elite climbers tend to use more their aerobic energy system and less their alactic anaerobic energy system than recreational climbers. According to [Guidi 1999], the three exercise energetic systems, coming from athletics studies performed on runners, are not sufficient to fully explain climbing performance, because a climbing effort is different from a running effort. The effort encounter in climbing is an intermittent effort, consisting of 70% of observation phases, and 30% of effective motion [Dupuy 1989].

In computer graphics and biomechanics The metabolic energy expenditure used in [Mordatch 2013] to simulate a character walking and running, first presented in [Anderson 1999], relates the force generated by a muscle with the corresponding needed thermal and mechanical energy. This model applies on Hill-type musculotendon units. It consists of four terms which are the muscle activation heat rate, the muscle maintenance heat rate, the muscle shortening heat rate, and the positive mechanical work rate.

Our energetic model and model comparisons Our energetic model is expressed at joint level, and the energetic cost consists in squared joint torques inte-

grated over time. As this model does not depend on joint positions nor joint velocities, it is comparable to the activation and maintenance heat rate of the model describe in the previous paragraph. These three energetic models, used in sport science, in graphics and robotics and ours, are different and complementary, as they operate at different observation levels. Indeed, human movements observed on an energetics point of view, can be seen as a chain of transformations, starting with the exercise energy systems (the first presented model) to convert food energy into muscular energy, continuing with the metabolic energy expenditure model to convert muscular energy into muscular forces, from which the joint torques can be determined, and finally involving the joint level model to convert joint torques into body motion. In this chain, we chose to be as close as possible to the body motion in order to reduce the model complexity, and thus we worked at joint level.

Chapter 2

Climbing Data Capture

Contents

2.1 Related Work **13**

2.2 Bouldering Data Capture **16**

2.2.1 Hardware and softwares, synchronization 16

2.2.2 Frames, wrenches, and their manipulation 25

2.2.3 Sensor frame calibration 32

2.2.4 Capture sessions 37

2.3 Top-Rope 2D-Video Capture **42**

2.3.1 Capture protocol 42

2.3.2 3D reconstruction of wall 42

2.1 Related Work

Several works have already studied rock climbing. Most of them aimed at exploring the physiological components of the performance in rock climbing. We review here those studies, and mention the device they used to collect climbing data.

Magiera et al. [Magiera 2013] date the first research interests in rock climbing in the late 1970s. A decade and a half later, Rougier and Blanchi [Rougier 1992] used a vertical wall with four load cells measuring 1-D forces, to measure the maximal voluntary contraction (MVC) a climber can generate. The MVC is defined as the maximal force that can be generated voluntarily on a given agonist muscle group. In [Rougier 1992], the MVC was approximated by using force measurements at the contact points between the climber and the wall. They conclude that the MVC is positively correlated with the expertise level of the climber, with a higher probability when taking the MVC normalized by the climber's weight. Three other studies [Testa 1999, Quaine 1999, Testa 2003] used the same setup, but with 3-D force sensors, to assess the anticipatory postural adjustement in response to the task of reaching a new hold from a quadrupedal posture. The last of those works also shows that the forces involved in the posture adjustment vary with the age of the climber.

The large space needed to practice rock climbing is an issue for in-laboratory studies of this activity. A workaround has been found by Booth et al. [Booth 1999], who used a vertical climbing treadmill to measure the physiological response to a

climbing effort. The physiology of climbing has also interested Mermier et al. [Mermier 2000], who showed the importance of training for rock climbing performance, and claims that the climbing excellence is not directly related to anthropometric characteristics. This claim has been discussed by Watts [Watts 2004] by gathering 12 existing works on rock climbing physiology to deduct the anthropometric and physiological performance factors, including that best climbers tend to be small with a low body fat level. The same year, the work by Sheel [Sheel 2004] has showed that the aerobic body capacity is a performance factor. Another interesting conclusion of Sheel is the impact of a psychological factor on the physiological response to rock climbing efforts. Finally, the unique work of de Geuss et al. [de Geus 2006] shows that several routes of the same difficulty can have different physiological impacts on the climber, depending on the style of the route. The technological innovation associated to this work is a continuous measurement of physiological data while climbing, enabled by portable devices.

An indoor climbing wall that can rotate has been used in a surgical context by Schoeffl et al. [Schoeffl 2004]. In this work, they mesured the pressure in the forearm muscles during climbing effort with a slit catherer. The aim was to evaluate the dangerousness of that sport on the forearm muscles. Fortunately, climbing is safe for the forearm muscles.

Sibella et al. [Sibella 2007] studied indoor climbing with a mocap system consisting of 6 infared cameras and 12 markers. Their conclusions, based on the assessment of the center of mass (CoM) motions, are that expert climbers minimize power during their ascent. A couple of works also studied climbing movements by capturing them with accelerometers [Schmid 2007, Pansiot 2008, Ladha 2013]. Accelerometers have the advantage of being lighter to set up than an infrared-based capture system, and can thus be used in a wider range of environments. The drawback of such systems is the absence of absolute reference frame. Thus they are more sensitive to drift.

3D force sensors mounted on climbing holds have been used in several works [Fuss 2006, Fuss 2008a, Fuss 2008b]. The first of these works interestingly concluded that powder chalk is far better than liquid chalk or a dry hand on clean surfaces, and that on messy surfaces, a dry hand is better than a powder-chalked hand. The two last studies are about the measure of contact forces applied on a hold during a competition. With 90% confidence they concluded that the competitor ranks are correlated to the Hausdorff dimension of the contact force distribution, this dimension summarizing the time of contact, that must be short, the amplitude of the contact force, that must be small, the good use of contact frictions, and the smoothness of the contact force through time. Lechner et al. [Lechner 2013] confirmed that result, with the breakthrough to have wireless 3D force sensors.

Finally, one of the most advanced works on the performance factors in rock climbing has been done by Magiera et al. [Magiera 2013]. In this work a correlation is established between the climber's peformance capacity, evaluated by the

Works	Climbers	Force Sensors	Physio	MoCap
[Booth 1999]	7	none	yes	no
[Mermier 2000]	44	none	yes	no
[Schoeffl 2004]	10	none	yes	no
[Watts 2004]	0 (review)	none	yes	no
[Sheel 2004]	0 (review)	none	yes	no
[de Geus 2006]	15	none	yes	no
[Magiera 2013]	30	none	yes	no
[Schmid 2007]	3	none	no	10 accelerometers
[Pansiot 2008]	4	none	no	1 accelerometer
[Ladha 2013]	53	none	no	1 accelerometer
[Rougier 1992]	17	4 sensors (1D)	no	no
[Testa 1999]	5	4 sensors (3D)	no	no
[Quaine 1999]	6	4 sensors (3D)	no	no
[Testa 2003]	32	4 sensors (3D)	no	no
[Sibella 2007]	12	none	no	6 cameras 12 markers
[Fuss 2006] [Fuss 2008a] [Fuss 2008b]	60	1 sensor type per session (4D)	no	no
[Lechner 2013]	21	1 wireless sensor (3D)	no	no
Our Work	9	6 sensors (6D)	no	24 cameras 36 markers

Table 2.1: *Characterization of the related works and our positionning with respect to them. The column Physio refers to some physiological quantity measurements, such as oxygen uptake, blood pressure, or heart rate for instance. The accelerometers measure the acceleration in 3D. Camera and marker devices involve infra-red cameras with passive markers.*

max on sight[1], and the max red point[2], with physical, technical and mental characteristics. Those characteristics include the finger strength, mental endurance (evaluated with psychological tests), climbing technique, the reaction time, the ape index and the oxygen uptake at anaerobic threshold.

The various features of the related works on climbing data capture presented above are summarized in the table 2.1. Our positionning with respect to the previous works is to capture both full 3D body motions thanks to 24 infared cameras and 36 markers, that we synchronize with six 6D force sensors mounted between the holds

[1]max on sight: hardest climb succeded on the first try, without any information on the ascent before the attempt. The difficulty is quantified with rating systems presented in table 2.2.

[2]max red point: hardest climb succeded after several tries, with eventually information on the ascent given by someone else.

and the wall, allowing us to have all the external contacts involved in the recorded climbing runs. The second capture experiment presented in the following has not such an excessive breakthrough, but is presented here as a simple and thus highly reproducible data set.

2.2 Bouldering Data Capture

Bouldering is a type of climbing where no harness is needed due to the low height of the wall. Usually the wall is about 3 meters high, which allows the climber to land safely on pads in case of fall. The small size of the wall needed and the fact that neither rope nor harness are needed makes this discipline a good technical choice for the in-laboratory study of climbing motions.

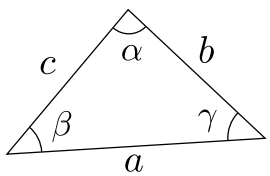
2.2.1 Hardware and softwares, synchronization

This sub-section describes the devices used to collect the bouldering data. Those data consist of both the 3D motion of the climber coupled with 6D measurements of the external contact forces and torques used along the motion.

Wall structure

The climbing wall structure on which the hold sensors are mounted (figure 2.1) has been build by an expert carpenter. The climbing surface is a square of 2.42 meters by 2.42 meters.

The wall has a negative slope of about four degrees. The slope has been computed from the side lengths of the external triangles of the structure. Given the following triangle



with known side lengths a, b and c, such that $a+b>c$, $a+c>b$ and $b+c>a$, i.e. the triangle is not flat, the relations to convert the side lengths into their respective opposite angles α, β and γ are given by

$$\alpha = \arccos\left(\frac{b^2+c^2-a^2}{2\ b\ c}\right) \tag{2.1}$$

$$\beta = \arccos\left(\frac{a^2+c^2-b^2}{2\ a\ c}\right) \tag{2.2}$$

$$\gamma = \pi-\alpha-\beta\ , \tag{2.3}$$

which is an application of the law of cosines. The external triangle side lengths have been measured between the three bolts that maintain this part of the wall structure.

Figure 2.1: *The instrumented bouldering wall used for the data capture. The holds are mounted on 6D force sensors. On each side of the wall one can see the tripods on which four of the 24 OtpiTrack cameras are mounted.*

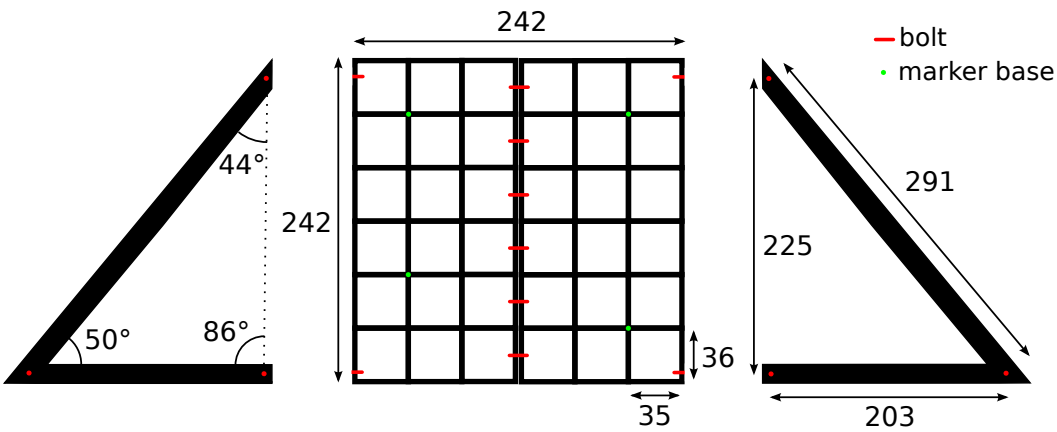


Figure 2.2: *The wall dimensions. All the lengths are in centimeters.*

These lengths are reported on the right side of the figure 2.2. With equations 2.1, 2.2 and 2.3, we computed the angles reported on the left side of the figure. Assuming that the bolts are centered on the central axis of the wood pieces they belong to, and that the lower wood beam of each triangle is horizontal, we conclude that the wall has a slope of $-4°$.

The central climbing surface consists of a torsion box forming a chessboard of 6 by 6 cells. The cells of 36 cm high and 35 cm wide can be filled with square wooden plank on which the holds and sensors are attached. Each of those boards is linked to the main structure by 8 bolts with butterfly nuts. This system of binding implies that two people are needed to change the configuration of the boards, one person in front of the wall and one person behind the wall simultaneously. Moreover, to change the 6 boards we have, 48 bolds need to be moved, which takes around one hour, if we count the alignement issues. Therefore, only two board configurations have been used for our data collection. The first configuration can be seen in the figure 2.1, the second configuration is shown in the figure 2.4.

Along with the wall structure, a set of 34 artificial Metolius climbing holds of diverse shapes is available. The holds are attached to the wooden hold support by a central screw, which does not collide with the screws that attach the hold support to the sensor. Indeed those last screws forms a circle with an empty interior.

Force sensors

The wall is equiped with six 6D force and torque sensors, from A-Tech Instruments Ltd. They are mounted between two wooden planks, as shown on the figure 2.3.

The size of the back sensor support and the size of the front hold support are the same for all the hold-sensor structures. The screw hole positions on those supports are also the same. By this way, the sensors can be placed in any wall cell. They can also be exchanged each other, while keeping the same location for the hold center location. We will further use this invariant for the calibration.

The sensor supports are rigidly attached to the wall framework. The pattern

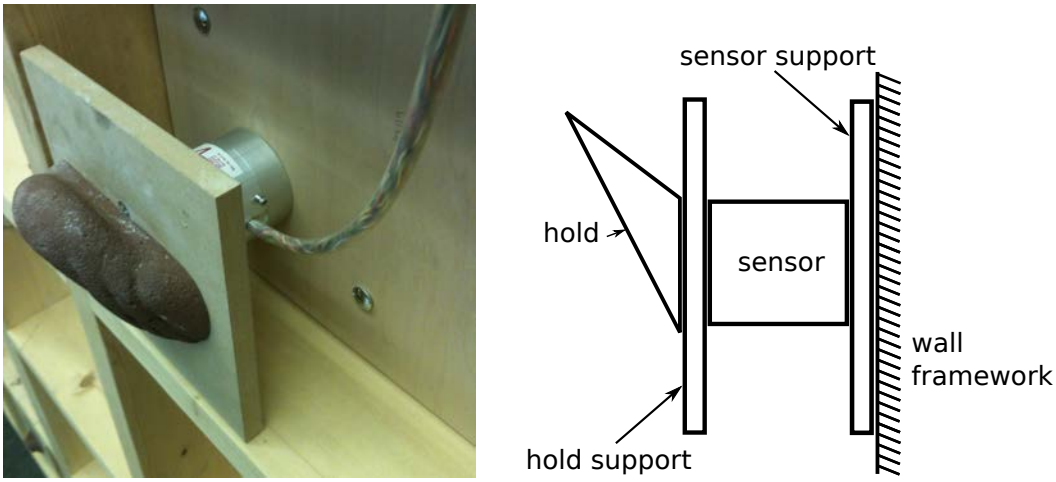


Figure 2.3: *A hold mounted on one of the six force and torque sensors (left), and its corresponding diagram (right).*

of the bolts allow the support to be rotated of any number of quarter turns. The sensor position on its support being ex-centered, the rotations are used to place the sensor in any cell corner. The hold supports are designed such that they can not touch the cell borders when the sensors are in place. The in-between space is about three millimeters.

The sensors are connected to amplifier boards, namely two AMTI MSA-6 Mini-Amp's, which allow us to set up the sensitivity of the measurements. As there are four sensor inputs per board, two boards are needed to connect the six sensors. Those boards are themselves connected to a computer via an Analog Input board, by CONTEC Co., Ltd. The data logger software is C-LOGGER, by the same company. This whole device can capture data at 1000 fps, but to simplify the synchronization process, we captured data at the same frequency as the motion capture device, which is 100 fps.

Mocap system

Motion capture, or mocap, is a standard way to capture 3D motions. The principle is that a set of several cameras emits and receives infrared light that reflects on small spherical markers, placed on the studied subject. After having recovered the 3D marker positions by triangulation, a skeletal model of the subject can be fitted on the marker trajectories. We detail in this section, the characteristics of the mocap device we used, the marker placements, the skeletal model, some calibration issues and finally some camera positioning issues.

An OptiTrack motion capture device with twenty-four V100:R2 cameras has been used to capture the climber's motions. The camera resolution is 640×480, and they capture data at 100 fps. Two of those 24 cameras can record low-quality

Figure 2.4: *The view from one of the two OptiTrack 2D video greyscale cameras.*

2D grayscale videos. An example of those videos is shown in the figure 2.4.

Several marker sets have been tested for the mocap, following the finite number choices provided by the OptiTrack Arena software to build the skeletal model to be captured. The marker configuration retained is exposed in the figure 2.5. It consists of 36 markers, associated with a skeleton of 18 rigid segments, and thus 17 joints. The skeleton model is presented in the figure 2.6. The grey ellipsoids are a visualization of the inertia matrices. Given the mass m_b of a rigid body, and the corresponding inertia matrix I_b in body frame, the ellipsoid semiaxes r_x, r_y and r_z are computed by inversing the system

$$I_b = \frac{m_b}{5} \begin{bmatrix} r_y^2 + r_z^2 & 0 & 0 \\ 0 & r_x^2 + r_z^2 & 0 \\ 0 & 0 & r_x^2 + r_y^2 \end{bmatrix}. \tag{2.4}$$

The charater mass distribution and inertial properties have been taken from the OpenSim models [Delp 2007]. Those quantities are scale for each climber according to its total mass.

The spine model includes three joints, located at the top of the spine, at the bottom of the neck, and at the lumbar vertebrae. The software allows a post-processing of the marker trajectories, in order to correct the automatic marker labeling, needed for the fitting of the skeleton on the marker trajectories. The full

checking of this labeling and its corrections for all the markers for a sequence of 30 seconds takes around 1 hour, when the correction is possible. To avoid a major increase of this post-processing time, neither the foot ball motions nor the finger motions have been captured.

The calibration of the mocap cameras is done with a wand on which three markers are rigidly attached. The TrackingTools software guides the process. The calibration process can take up to two hours due to the slow convergence of the calibration algorithm. Sometimes, the wand sweeping had to be done several times because of the non-convergence of the calibration optimization. We have empirically noticed that the quality of the calibration is improved by moving the wand slower, probably due to camera blur. The camera calibration process must be done each time a camera is moved. As the tripod cameras that are in front of the capture room door (see figure 2.1) are moved after each capture day, the calibration process has been done in the morning of each capture day.

Those tripod cameras on each side of the wall has been placed in order to see the markers between the climber and the wall. Two other cameras on the same plan have been placed 50 cm above the wall. But for some captured motions, those six side cameras appeared not to be enough. From time to time, we observed the disappearance of some markers, leading to an erroneous skeletal motion reconstruction, as discussed further in 2.2.4.

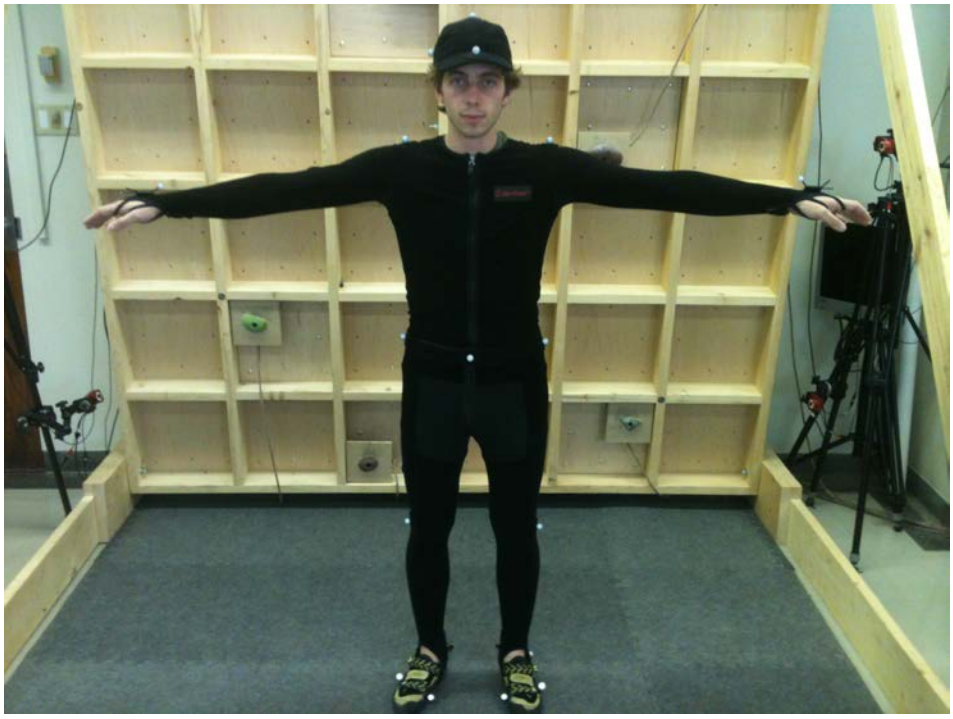
Sensor and mocap synchronization

In order to perform a dynamical analysis of the captured motions, the force recording devices must be synchronized with the mocap system. Two synchronizations must be done, one per amplifier board. The method chosen is to hit two hold supports, connected to the two amplifier boards, at the beginning of each recorded sequence. This hitting is done by the climber, such that the hitting motion is recorded by the mocap system. The hit instant is then visually located on the corresponding force curves (in C-LOGGER), and on the hand marker trajectories (in Arena). With these two instants we can match the force measurement with the motion measurement.

This visual processing is ploted in the figure 2.7. Its precision is estimated about 10 ms, which is the precision of the motion (see figure 2.7b). With respect to this precision we can neglect the precision lost in the downsizing of the sensor sampling rate from 1000 fps to 100 fps. Therefore, we used this downsizing to reduce the number of post-processing steps.

Another solution for data synchronization would have been a hardware synchronization. This solution has not been retained, due to the heterogeneous set of devices used in the data capture.

Having presented the hardware and softwares used to capture the motions needed for our study, and how their temporal synchronization is done, we will now move to some more theoretical concepts needed to spatially calibrate the mocap system with the force sensors.



(a) *Frontside view of the marker set.*

(b) *Backside view of the marker set.*

Figure 2.5: *The location of the 36 markers used for the motion capture. The four markers on each side of the ball of the feet are mounted on a hard plastic base, attached to the climbing shoes via double face tape. The other markers are attached to the motion capture suit with a hard Velcro base for the six hand markers and a soft Velcro base for the other markers.*

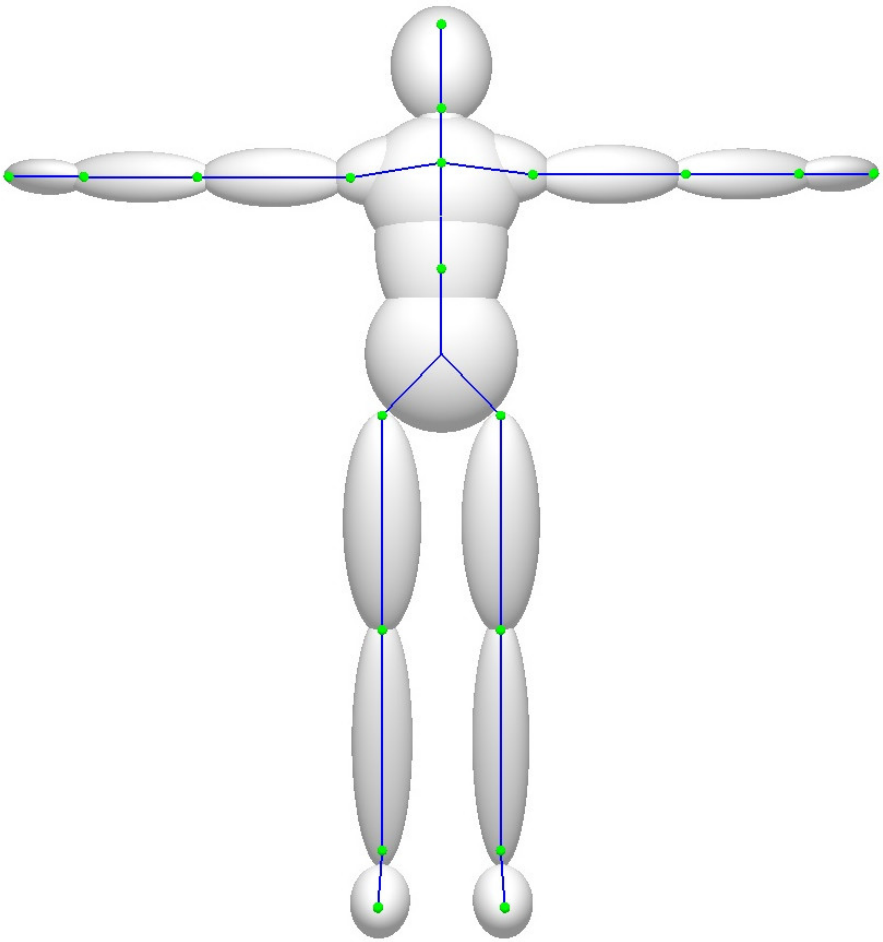
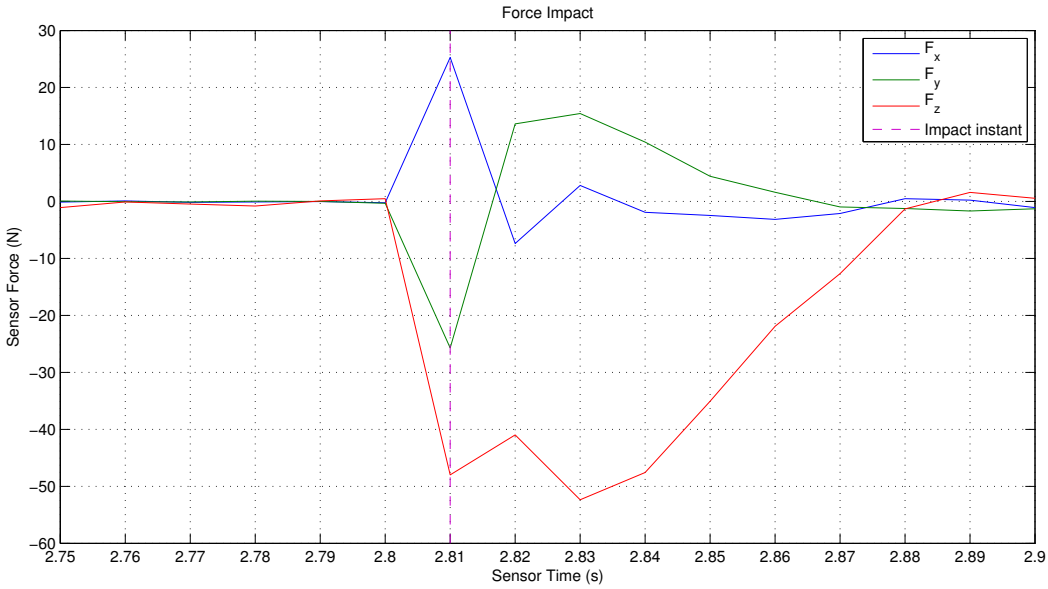
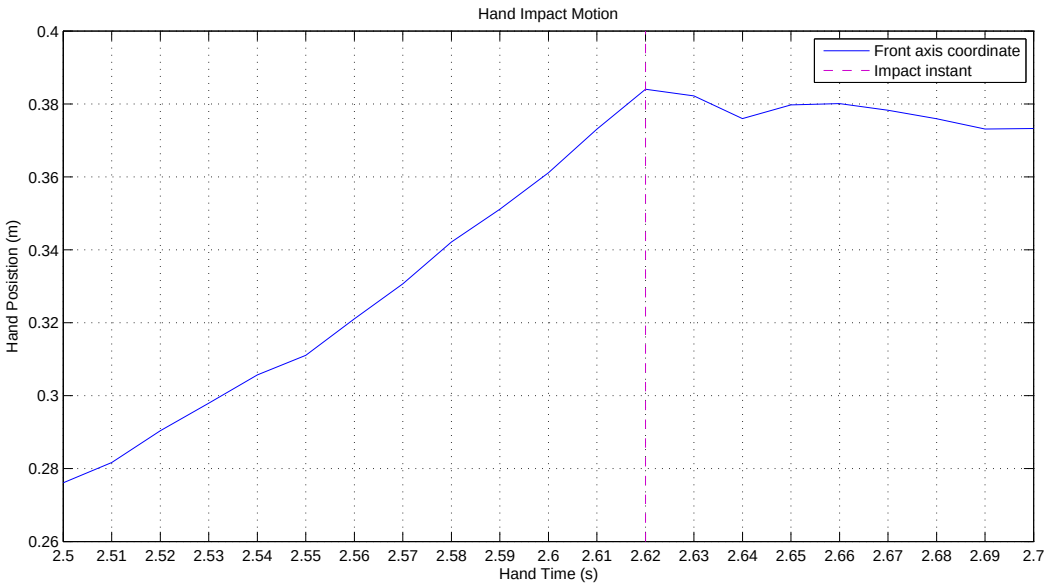


Figure 2.6: *The skeletal model used for the motion capture. The fitting of the marker trajectories by inverse kinematics is done internally by OptiTrack. The OptiTrack output does not include 3-D marker positions. All joints are ball joints.*



(a) *Synchronization impact on the sensor point of view, before synchronization.*



(b) *Synchronization impact on the hand point of view, before synchronization.*

Figure 2.7: *The synchronization process. The impact is located with both (a) the forces measured by the sensor, and (b) the motion of the hand.*

2.2.2 Frames, wrenches, and their manipulation

Frames, wrenches and the mathematical operators for their manipulate, namely translations, rotations, and adjoints, constitute the core of this thesis. Complete mathematical definitions exist in [Murray 1994], as well as efficient implementations (e.g. [Guennebaud 2010, Liu 2013]). Consequently, we focus here on providing the intuition associated with those notions.

Frames and transformations

A frame, or coordinate frame, is defined by its origin and orientation. The origin is a 3D point, and the orientation is a basis made of 3 orthonormal vectors. Those vectors can be gathered to form the columns of a rotation matrix. Origin and orientation are themselves defined in a frame. All frames are thus defined with respect ot each others. We call the frame with origin $(0, 0, 0)$ and with orientation the unit basis vectors the world frame. By default all frames are defined with respect to a world frame. Coordinates expressed in that frame are called world coordinates, global coordinates or maximal coordinates. All other frames are called local frames, and coordinates expressed in those frames are called local coordinates.

The rotation is to the orientation, what the translation is to the position. To rotate an object, one needs an axis along which the rotation is performed, and an angle to define the amplitude of the rotation. In practice, those two quantities can be stored in one 3D vector by multiplying the axis (a normalized vector) by the angle. The advantage of this representation is that each angle-axis represents a unique rotation, whereas rotation matrices or quaternions can represent an infinite number of rotations. Indeed, the difference between a rotation of two turns and no rotation, i.e. a rotation of $0°$, can be made with angle-axes, but with quaternions, those two rotations are exactly the same. Moreover with the angle-axis representation, angular velocities are directly defined by rotations (represented by 3D angle-axes, and not by 3-by-3 matrices) divided by the time interval between the original and final orientations. The three components of the vector defining such axis-angle representation are called the canonical coordinates, or the exponential map coordinates of a rotation.

Given two orientations, i.e. two rotation matrices, only one rotation is needed to transform the first orientation into the second one (Euler's Theorem). We will illustrate this important theorem by showing how such a rotation can be constructed geometrically. A mathematical proof of this theorem can be found in [Murray 1994]. Let's considere the rotation of a vector from one position into another, as the two black vectors of the figure 2.8. We want to find the angle-axis corresponding to this rotation. As Rotations preserve angles, the rotation axis must forms equal angles with the initial and final vectors. Thus all possible rotations belongs to the median plan of these two vectors. Let's move to the rotation of frames, as illustrated by figure 2.9. By applying the previous reasonning, the rotation axis is found at the intersection of the three median plans of the pairs of basis vector. The angle is then

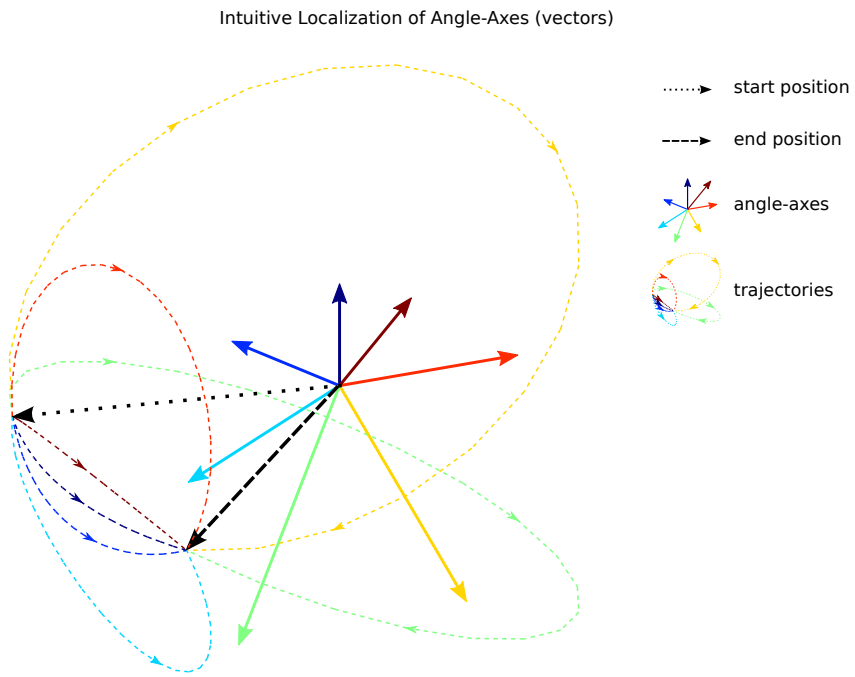


Figure 2.8: *Seven possible rotations for transforming a vector from a start position (dotted black vector) to a end position (dashed black vector). All the angle-axes belong to the median plan of those two vector positions.*

obtained by projecting the bases in the plan orthogonal to the axis. This process provides an intuitive localization of the angle-axis rotating two given frames.

A transformation, or a frame transformation, is an opperator that converts coordinates from one frame into another frame. From the Euler's theorem, any frame transformation can be expressed with one translation and one rotation.

We have finally presented the spatial notions we will use to describe the motion of any rigid body, namely frames and transformations. We will now present the notions needed to describe the dynamical interaction between objects.

Forces and torques

The notion of force can be felt by holding a mass object in the hand, and observing the action of this object on the palm of the hand. A force is also what is exerted on shoulders when carying a heavy backpack. It is also what the floor applies on the feet when walking. A force has a point of application, a direction and a magnitude, so we represent forces by vectors.

The torque is to the force, what the rotation is to the translation. There are as much differences and similarities between the notion of torques and forces as there are between rotations and translations. A torque is what a car wheel received to turn. A torque is what is felt at the knees when doing squat down motions, or at

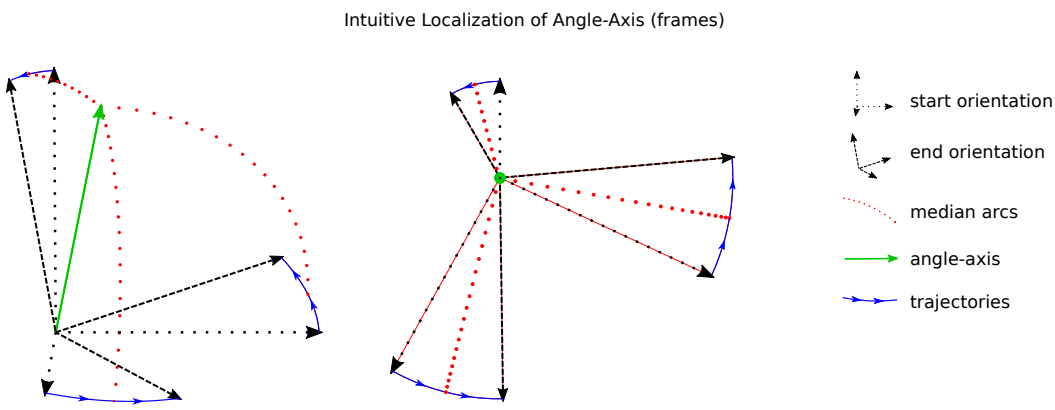


Figure 2.9: *Angle-axis geometric construction for frame rotations. The axis (green) is at the intersection of the median plans of the pair of basis vectors (red dotted arcs intersection). The angle is found in the orthogonal plan to the green vector (middle diagram).*

the shoulders when carying heavy shopping bags. We represent torques by vectors. Both torque vectors and rotation vectors (angle-axes) follows the same convention : turning clockwise the vector goes forward, turning counterclockwise the vector goes backward. The convention is illustrated by the angle-axes plotted in the figure 2.8 with their associated trajectories.

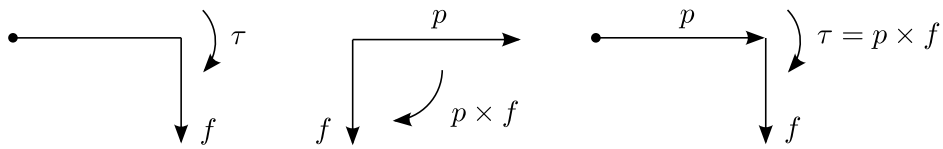
Torques sum up at joints the muscle actuation. All the linear forces transmitted by the muscles around a joint can be gathered into a single torque, which simplifies the reasonning. The similarity with rotations is obtained when considering the translation of several points of a wheel, that can be more compactly expressed with the wheel rotation. Joint torques are the coarser level for studying the human motion actuation[3]. This level is therefore more convenient to manipulate than the muscle actuation level.

Forces and torques are related by lever arms. This relation can be written by the cross product

$$\tau = p \times f \tag{2.5}$$

where f is the force producing the torque τ by the lever arm p. p is the vector from the point of application of τ to the point of application of f. The direction of p can be determining with the visual process proposed in the figure 2.10. First, the direction of τ is determined by thinking of how the arm would turn when applying the force f (see figure 2.10a). Second, the cross product direction is determined by saying that we go from the left side argument of the cross product towards the right side argument, when the two vectors start from the same point like two clock hands (see figure 2.10b). Finally, the relation between those two directions is established (see figure 2.10c).

[3]human motion actuation are the internal forces and torques involved in the human motion



(a) *τ direction* **(b)** *p direction* **(c)** *lever arm relation*

Figure 2.10: *Three visual steps to establish the lever arm torque-force relation.*

Now that forces and torques have been presented, we can combine them to form wrenches. Wrenches are transfered from frames to frames by adjoints.

Wrenches and adjoints

Wrenches and adjoints are respectively elements and operators evolving in $\mathbb{R}^6$. By using real-life examples of wrenches, we aim to make this abstract mathematical space more intuitive for physical applications.

A wrench is the combinaison of a force and a torque. To set the notations, let's $w \in \mathbb{R}^6$ be a wrench. The force f and the torque τ associated to this wrench will be arbitrarily written in the order

$$w = \begin{bmatrix} f \\ \tau \end{bmatrix}. \tag{2.6}$$

f is the linear part or the linear components of the wrench. τ is the angular part. τ is also called a moment. The difference between the terms torque and moment is that a moment is the angular components of a wrench, whereas a torque is a wrench with a null linear part. We will represent wrenches by the composition of straight vectors for the linear part, and arc vectors for the angular part.

To illustrate the concept of wrench, let's consider we are trying to drive a screw into a flat plank of wood using a manual screwdriver, without pre-drilling the hole. Applying only a torque would lead to make the screw turning on itself, without diving into the wood. On the contrary, applying only a force can only lead to nailing the woodscrew into the wooden plank, and not driving it. In that case, the quantity that our hand must apply to the screwdriver in order to obtain a clean driving, is what we called a wrench.

Wrenches can be transfered from one point of application to other points of application via rigid objects. This transfer affects the value of the wrench, but not its global effect. In other words, wrenches are location dependent and wrench transfer is conservative. To illustrate the notion of wrench transfer, we replace the screwdriver of the previous example by an Allen key (see figure 2.11). When placing our hand on the longest side of the key, we change the point of application of the wrench provided by our hand on the tool, as compared with the point of application obtained with the screwdriver. But the wrench received by the woodscrew can be identical as before if we change the wrench we generate at our hand location, and

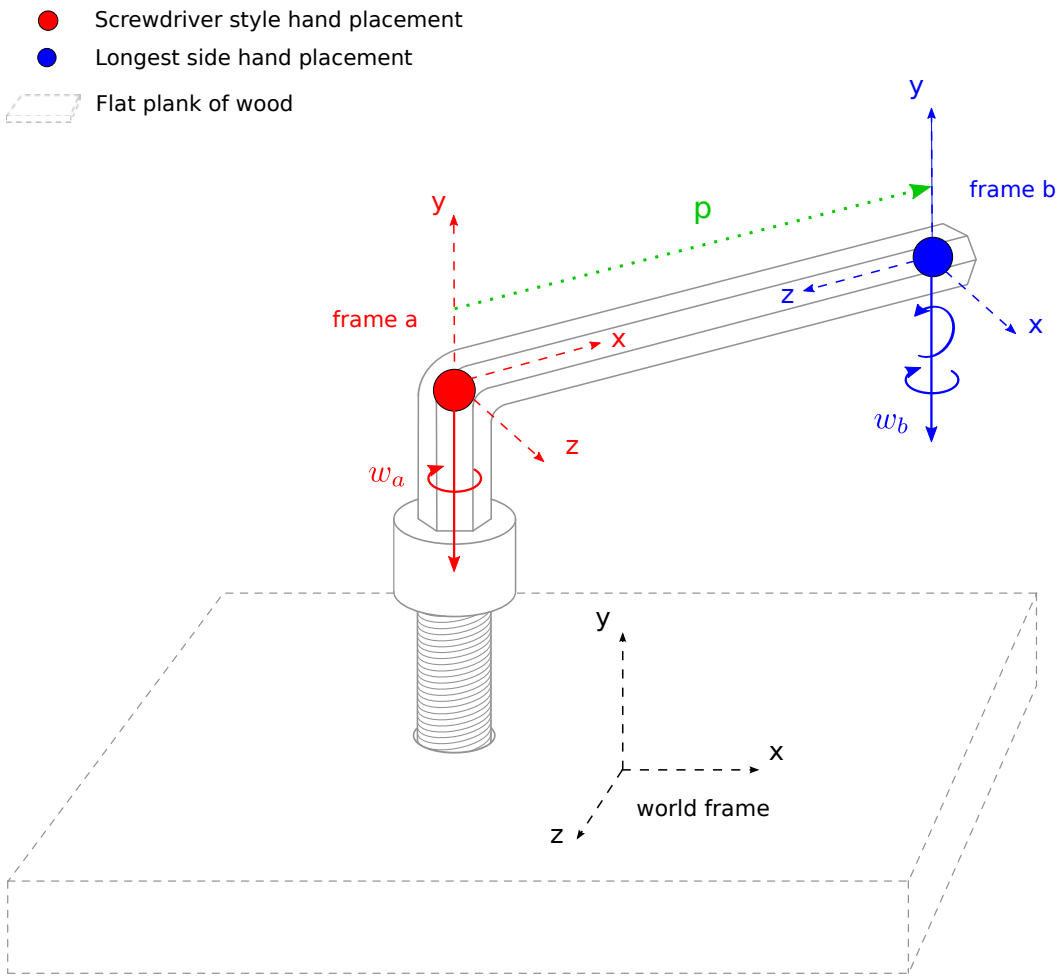


Figure 2.11: *Wrench transfer illustration with an allen key. The wrenches w_a (red) and w_b (blue) applied at two different points of the Allen key (large dots), have the same effect on the screw. For w_b, a torque component on the x-axis local axis is added to compensate for the torque resulting from the lever arm relation (see figure 2.10c), when transfering w_b from frame b to frame a.*

add some additional angular components. The new wrench is thus tranfered from our hand to the screw head through the rigid Allen key. The intuition of the effect of this transfer can be obtained by considering the two wrenches that we provide when holding the two different tools. The two provided wrenches are different, but result in the same effect on the screw head. The first wrench can thus be seen has the result of the transfer of the second wrench to a new location, and vice versa.

Definition The adjoint is the mathematical operator that transfers a wrench from one frame into another frame. If the frames are noted a and b, then this transfer can be written

$$w_a = \mathrm{Ad}_{ab}^T \, w_b \ , \tag{2.7}$$

where w_a and w_b are the transfered wrench observed at frames a and b respectively, and Ad_{ab}^T is the adjoint matrix transfering the wrench from b to a. To build this adjoint matrix, we first assume that the frames a and b have the same orientation. In that configuration, the new wrench is obtained be adding to the former wrench the torque coming from the lever arm equation 2.5. We obtain

$$w_a = \begin{bmatrix} I & 0 \\ p^\times & I \end{bmatrix} w_b \ , \tag{2.8}$$

with I the 3-by-3 identity matrix, 0 the 3-by-3 zero matrix, and $p^\times$ the cross product matrix. Noting $p = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T$, this last matrix is defined by

$$p^\times = \begin{bmatrix} 0 & -p_z & p_y \\ p_z & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix} . \tag{2.9}$$

The direction of p is ruled by the lever arm definition. By using the figure 2.10, we can figure out that p is the vector from the origin of frame a towards the origine of frame b. Finally when the orientations are varying from frames to frames, we need consider the frame in which p is expressed. We will label it frame c. In order to applied equation 2.8, wrenches must be express with frames having the same orientation. We thus add on each side of the adjoint matrix of equation 2.8, the rotation matrices from b to c and from c to a. The complete adjoint definition is then

$$\mathrm{Ad}_{ab}^T = \begin{bmatrix} R_{ab} & 0 \\ R_{ac} \, p^\times \, R_{cb} & R_{ab} \end{bmatrix} , \tag{2.10}$$

where R_{cb} is the rotation from the frame b to the frame c, in which p is expressed, R_{ac} is the rotation from frame c to frame a, and

$$R_{ab} = R_{ac} \, R_{cb}$$

is the rotation from frame b to frame a.

The transpose notation for the adjoint comes from the original adjoint definition, which established, similarly to wrenches, that linear and angular velocities can be transfered from a frame a to a frame b by the relation

$$v_b = \mathrm{Ad}_{ab}\, v_a \ , \tag{2.11}$$

with v_a and v_b the 6D vector with the linear and angular velocities of a solid, observed at frame a and frame b respectively.

Example Let's write the wrench transfer equation 2.7 for the Allen key example (figure 2.11). The rotation to change the world frame orientation into the frame a orientation is a rotation of $\alpha = 30°$ around the y axis. This is also the rotation to go from the local coordinates of frame a, to the world coordinates. Similarly, the frame b orientation is a $\beta = -60°$ rotation about the y-axis. These two rotations will be noted

$$R_a = \begin{bmatrix} c_\alpha & 0 & s_\alpha \\ 0 & 1 & 0 \\ -s_\alpha & 0 & c_\alpha \end{bmatrix}, \qquad R_b = \begin{bmatrix} c_\beta & 0 & s_\beta \\ 0 & 1 & 0 \\ -s_\beta & 0 & c_\beta \end{bmatrix}, \tag{2.12}$$

where c_α, s_α, c_β and s_β are respectively $\cos(\alpha)$, $\sin(\alpha)$, $\cos(\beta)$ and $\sin(\beta)$. We decide to write the vector p in the frame a, such that it has only a component on the x-axis, which gives

$$p_a = \begin{bmatrix} p_x^a \\ 0 \\ 0 \end{bmatrix}. \tag{2.13}$$

The wrenches w_a and w_b are also expressed in local frames to reduce their number of components, thus

$$w_a = \begin{bmatrix} 0 \\ f_y^a \\ 0 \\ 0 \\ \tau_y^a \\ 0 \end{bmatrix}, \qquad w_b = \begin{bmatrix} 0 \\ f_y^b \\ 0 \\ \tau_x^b \\ \tau_y^b \\ 0 \end{bmatrix}. \tag{2.14}$$

Let's now transfer w_b to w_a, corresponding to a wrench transfer from frame b to frame a, and find the condition for τ_x^b to vanish. In the equation 2.10 for our case, p is expressed in the frame a, so the frame c is the frame a : $R_{ac} = I$ and $R_{cb} = R_{ab}$. If $\gamma = -90°$, $c_\gamma = \cos(\gamma)$ and $s_\gamma = \sin(\gamma)$, we have

$$R_{ab} = \begin{bmatrix} c_\gamma & 0 & s_\gamma \\ 0 & 1 & 0 \\ -s_\gamma & 0 & c_\gamma \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}. \tag{2.15}$$

R_{ab} transforms local coordinates from frame b to frame a (the x-coordinate in b gives the z-coordinate in a). Conversely when looking at the figure 2.11, where all is drawn

in world coordinates, R_{ab} rotates the frame a into the frame b (counterclockwise rotation of 90°). Finally equation 2.7 in our example gives

$$\begin{bmatrix} 0 \\ f_y^a \\ 0 \\ 0 \\ \tau_y^a \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ -p_x^a & 0 & 0 & 0 & 1 & 0 \\ 0 & p_x^a & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ f_y^b \\ 0 \\ \tau_x^b \\ \tau_y^b \\ 0 \end{bmatrix}. \tag{2.16}$$

We obtain $f_y^a = f_y^b$ from the second line of this system, $\tau_y^a = \tau_y^b$ from the line 5, and from the last line, $\tau_x^b = -p_x^a f_y^b$, which is the condition to have no torque on the z-axis in the frame a.

2.2.3 Sensor frame calibration

Once frames, wrenches, and their associated manipulation quantities have been defined, we can use them to calibrate the force sensors. The calibration aims to determine a commun frame between the sensors and the mocap. This process has been described in [Aladdin 2012]. For the completness of this manuscript, and because we took part to some technical discussions, we summarize the geometric part of this process here, and add some complements.

Process overview

The main idea of the calibration process is to apply forces at a known location with no torques. In the figure 2.14, this point is the point C. The applied forces are measured by the sensor at point S with some torque components. Knowing the wrenches at the sensor, the tool center of pressure C, and assuming that there is no torque at point C, the point S is obtained with the wrench transfer equation. This process is detailed in the following two sections. The section "Tool center of pressure" will describe a tool equiped with motion capture markers (figure 2.12), which is rotated around its tip (the center of pressure) to find the tip coordinates by solving a least square system. The section "Sensor center of measure" will detail how to use the wrench transfer equation to retrieve the relative position of the sensor center of measure and the tool center of pressure.

Tool center of pressure

A center of pressure is where the wrench has no angular component in the plan tangential to its linear components. In the case of a ground contacts for instance, the center of pressure is uniquely defined as the intersection of the contact plan with the line where the angular part of the wrench applied by the foot onto the ground has only a component in the direction of the linear part of this wrench (see figure 2.13). For wrenches with no linear components, the center of pressure is thus not defined. As the tool tip consists of a sharp nail, we can consider that the apply wrenches

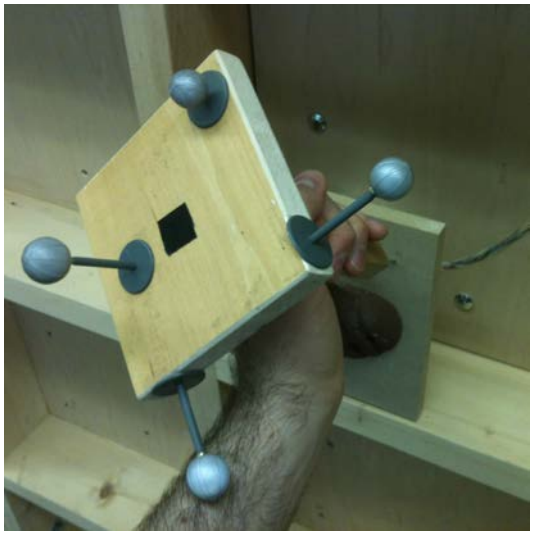


Figure 2.12: *The calibration tool is equiped with mocap markers to define the tool frame.*

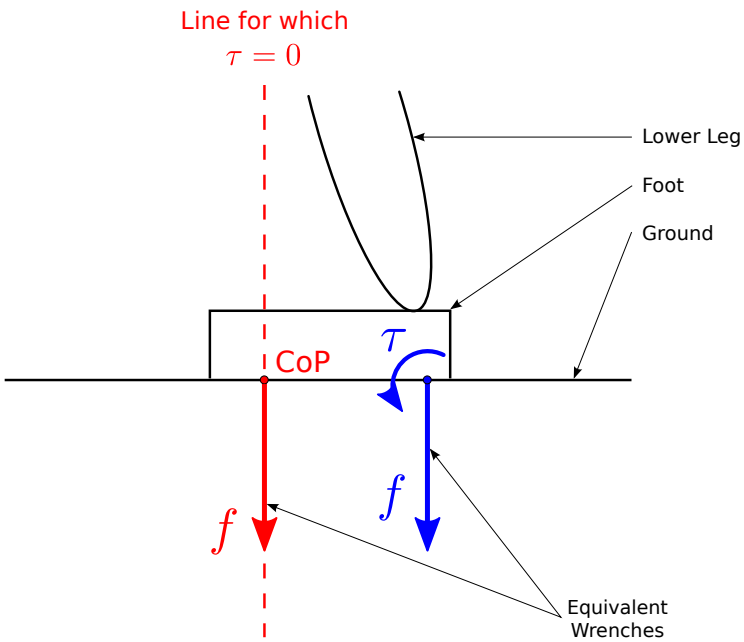


Figure 2.13: *Center of pressure (CoP) for a ground-foot contact. The blue wrench is the contact wrench, seen at a point below the heel. The red dashed line is the line where the wrench equivalent to the blue wrench has no torque. The CoP is at the intersection of this line and the ground.*

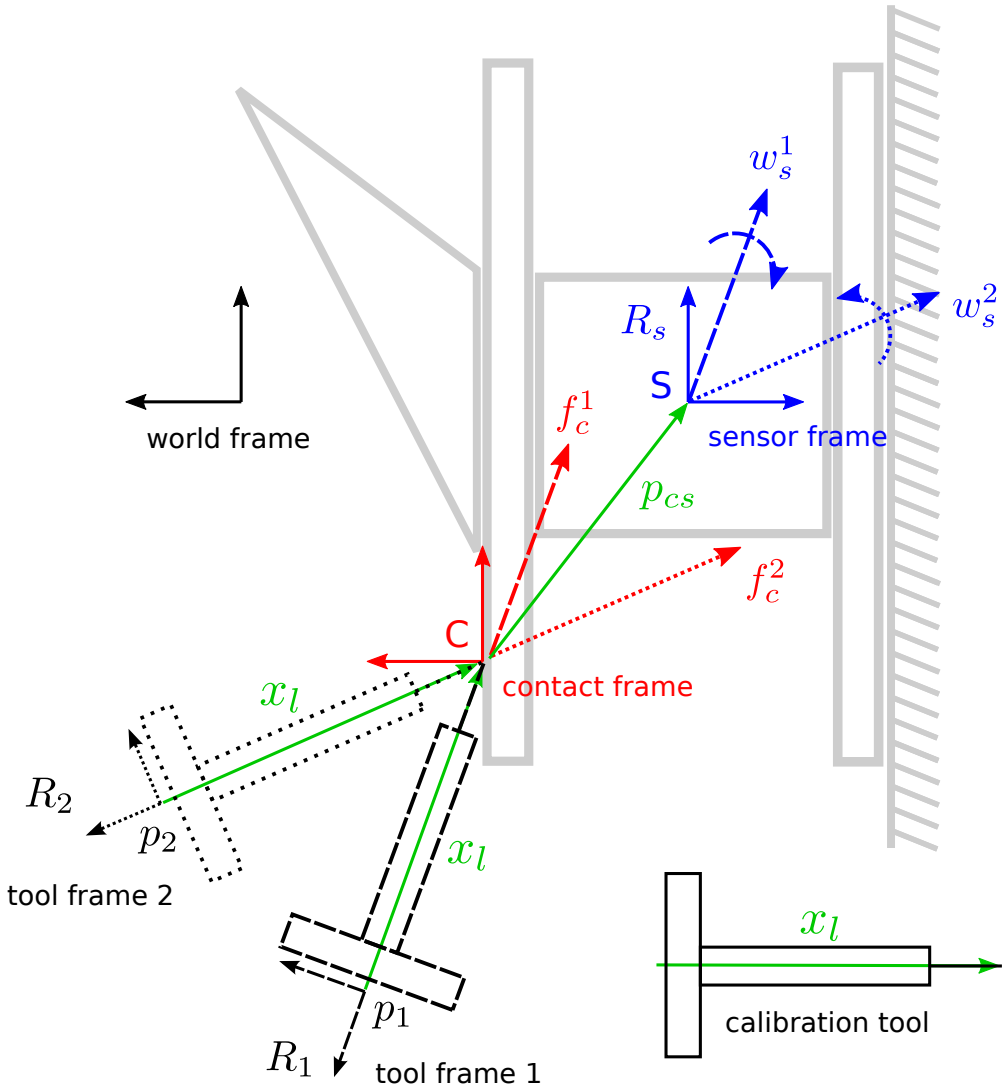


Figure 2.14: *The quantities used in the calibration process. C is the contact point of the tool tip, corresponding to the tool center of pressure. S is the sensor center of measure. x_l is the local tool tip coordinates.*

have no angular component at the tip contact point. The center of pressure will thus be the sharp extremity of the calibration tool.

To locate this center of pressure, n frames of the tool rotating around that point are recorded. The origin and the orientation of the tool are arbitrarily set with respect to the tool markers, and are locally consistent between all recorded frames. The world orientation of the frame j, and the world origin position of the frame j will be noted respectively

$$R_j,\ p_j,\ \forall j \in 1..n. \tag{2.17}$$

The aim is then to find a point of local tool coordinates x_l and world coordinates C verifying

$$R_j\ x_l + p_j = C,\ \forall j \in 1..n, \tag{2.18}$$

which is a linear system of $3n$ equations for 6 unknowns. This equation can be rearranged under the classical form

$$\begin{bmatrix} -R_1 & I \\ -R_2 & I \\ \vdots & \vdots \\ -R_n & I \end{bmatrix} \begin{bmatrix} x_l \\ C \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}, \tag{2.19}$$

Finally, taking a number of frames n greater than 2, we can solve this system in the least square sense.

Sensor center of measure

Knowing the world coordinates of the set of contact points C between the calibration tool and the hold support, as well as a corresponding set of wrenches apply by the tool, and measured at the sensor center, we can compute the sensor center S and its orientation R_s (see figure 2.14). To explain this calibration process, we start with the mathematically simplest case with only one contact point and one wrench, which is then incremented step by step to obtain the complete process.

Let's call s the sensor frame, and c the contact frame, with world aligned axes. The force f_c applied by the tool on C, and the wrench w_s measured by the sensor at S, are related by the wrench transfer equation 2.7, which gives us

$$\begin{bmatrix} f_c \\ 0 \end{bmatrix} = \mathrm{Ad}_{cs}^T\ w_s. \tag{2.20}$$

In this equation, Ad_{cs}^T is the adjoint converting wrenches from frame s to frame c. This adjoint depends on the vector from C to S, namely p_{cs}. It also depends on the orientation of frame s with respect to frame c, which is also R_s, the world orientation of frame s, due to the world axis alignment of the frame c.

The system 2.20 has 6 equations for 9 unknowns, that the variables f_c, p_{cs} and R_s consist of. Without losing information, the three first equations and the three first unknowns are removed, leading to the system

$$\begin{bmatrix} p_{cs}^{\times}\ R_s & R_s \end{bmatrix} w_s = 0. \tag{2.21}$$

To have more equations than unknowns, several wrenches w_s are recorded.

The use of a nonlinear solver is avoided by iteratively solve for p_{cs} with R_s fixed, and for R_s with p_{cs} fixed. The first step is a least square solving in p_{cs}. p_{cs} is isolated using $a^\times b = -b^\times a$. For the second step, we iteratively solve for small perturbations of the current R_s

$$R_s' = R_s \left(I + \omega^\times\right), \tag{2.22}$$

leading to a sequence of least square systems in ω. $(I + \omega^\times)$ is the first order approximation of a small rotation of exponential coordinates ω. The initialization of p_{cs} and R_s is done with a rough estimation, ensuring that the closest local minimum is the solution we are looking for.

Finally, note that if (p_{cs}, R_s) of the figure 2.14 is a solution for the equation 2.21, then for any rotation R, the pair $(R\ p_{cs}, RR_s)$ is also a solution. Indeed, starting from the left of equation 2.21 with this new candidate, as

$$(R\ p_{cs})^\times = R\ p_{cs}^\times\ R^T, \tag{2.23a}$$

we have

$$\left[\ (R\ p_{cs})^\times R\ R_s \quad R\ R_s\ \right] w_s = R \left[\ p_{cs}^\times\ R_s \quad R_s\ \right] w_s = 0\ , \tag{2.23b}$$

because (p_{cs}, R_s) verifies equation 2.21. Thus $(R\ p_{cs}, R\ R_s)$ is also a solution for equation 2.21. To overcome this singularity, we collect wrenches at three different points of contact, and add to the system 2.21 the known relative contact point positions in the world frame, expressed in term of p_{cs}'s. By this way, the global orientation of the points are constrained, leading to a unique solution for the sensor frame orientation.

To conclude, given a set of three points of known global coordinates, and several sensor measurements of wrenches produced by exercing linear wrenches at those points, the sensor center is found by solving a system of equation with the sensor orientation and the vectors from the contact points to the sensor center as unkwowns. The first part of this system consists of the transfer equation of the measured wrenches from the sensor center to the contact points, where the torque components are supposed to vanish. The second part is the equality of the relative positions of the contact points, expressed with the vectors from the contact points to the sensor center. This last part, not mentioned in [Aladdin 2012], imposes the global orientation of the contact points while solving for the cancellation of the angular components of the measured wrenches.

Sensor frames in the wall structure frame

The sensors frames are calibrated with respect to the wall structure by the intermediate use of the positions of the 6-by-6 cells in which the sensor supports are embedded in.

These intermediate quantities reduce the time needed to calibrate the sensors, at the cost of adding inaccuracies, brought by the irregularities of the wall building

process. Those imprecisions occure both on the orthoganality and on the spacing regularity of the cell borders, as well as on the regularity of the drilling pattern of the holes used to insert the bolts that maintain the sensor supports, and the holes to fix the sensors on their support. We estimate those imprecisions to be of a few millimeters, which is acceptable in comparison with the assumption that the body markers fastened to the mocap suit by Velcro supports, are rigidly attached to the climber's body segments.

The wall frame is calibrated with respected to the mocap system by using four markers placed on plastic supports glued to the wall structure. These supports are represented in green in the figure 2.2, and can be seen on the picture of the figure 2.1. The markers are placed after each calibration of the mocap device. They are removed for the mocap recordings, to avoid the climber to be disturb by the additional task of avoiding them.

The cell coordinates are recorded by measuring their top-left corner coordinates. The calibration tool is used to that end. The sensor calibration process previously described is then done inside a reference cell. The cell center is computed from the cell corners. The calibrated sensor support can then be rotated around the cell center in the wall plan by any multiple of 90°, as allowed by the regular 8 bolts pattern of the sensor support. It can then be translated to any other cell.

Therefore, knowing the calibration of the wall, thanks to the coordinates of four markers fixed once for all on the wall, the sensor frames are retrieved from the orientation of the sensor supports, and there discrete coordinates in the 6-by-6 cell chessboard.

2.2.4 Capture sessions

The study of climbing motions implies the participation of climbers, and the design of bouldering problems. This section describe the climbers involved in the capture sessions, the boldering problems that we have proposed to them, the collected data and their post-processing.

Climbers and bouldering problems

Nine climbers have participated to the data captures. Two beginners, two intermediate climbers, and five expert climbers. The beginners are men climbing for the first or for the second time. The intermediate climbers are women climbing between 5.10d and 5.11b. The expert climbers are men with best redpoint ascent from 5.13a to 5.14a. We will now explain this rating system.

Climbing routes and bouldering problems[4] can be classified according to their difficulty on a 1-dimensional rating scale. This single dimension can not reflect the whole diversity of all the climbing routes and bouldering problems. For instance, a

[4]Route climbing and bouldering are two different climbing disciplines. A route is about 10m or 20m high and both rope and harness are needed to climb. A boulder is about 3m high, thus no harness is needed, just a safety crash pad.

Free climbing		Bouldering		Need	Interpretation
YDS	Fra.	Hueco	Bleau		
3-4	1				
5.1	2				Hiking.
5.3	3				
5.4	4a			1 week	Ladder movements.
5.5	4b				
5.6	4c				
5.7	5a			1 month	Non-intuitive movements on good holds.
5.8	5b		3		
5.9	5c	V0-	4-		
5.10a	6a	V0	4	1 year	Medium holds with technical foot placements.
5.10b	6a+	V0+	4+		
5.10c	6b	V1	5		
5.10d	6b+	V2	5+		
5.11a	6c	V3	6A	2 years	Very small handholds, medium power movements.
5.11b	6c+	V4	6A+		
5.11c					
5.11d	7a	V5	6B		Short powerfull or technical overhanging sections.
5.12a	7a+		6B+	5 years	
5.12b	7b	V6	6C		
5.12c	7b+		7A		Long apnea sections, due to high muscle contraction.
5.12d	7c	V7	7A+	10 years	
5.13a	7c+	V8	7B		
5.13b	8a		7B+	15 years	Tiny holds for very long and specific movements.
5.13c	8a+	V9	7C		
5.13d	8b	V10	7C+		
5.14a	8b+	V11	8A	to be gifted	High precision movements at strength and mental limits.
5.14b	8c	V12	8A+		
5.14c	8c+	V13	8B		
5.14d	9a	V14	8B+	to be special	Physical and mental battle, with random success.
5.15a	9a+	V15	8C		
5.15b	9b	V16	8C+		
5.15c	9b+				

Table 2.2: *Climbing grade equivalence estimation between the free climbing Yosemite Decimal System (YDS), the free climbing french system (Fra.), the bouldering Hueco Tanks system (Hueco), the bouldering Fontainebleau system (Bleau), our estimation of their associated prerequisites (Need), and our interpretation of the grades. Free climbing ratings are established on lead. The provided times are the practice durations for someone training twice to three times a week. As no absolute rating scale exists, this table comes from several diverse sources, to which we add our own experience.*

Step	Description	Selected	
		runs	%
0	Initial raw data.	513	100
1	Recording failures.	480	94
2	Removing static poses and marker tunings.	292	57
3	Obvious failure visualization. (figure 2.15)	159	31
4	Thresholding joint velocity at 100 rad.s^{-1}.	83	16
5	Blackman filtering. (figure 2.16)	83	16
6	Marker noise detection.	30	6
7	Final qualitative selection.	15	3

Table 2.3: *McGill data selection process.*

powerfull and tall climber can be confortable with dynamical movements, also called dyno's, and stucks on problems with small movements but technical body positioning. On the contrary, a small and technical climber could have the exact opposite difficulty, while climbing the same grades. Another more concrete example is the work of de Geus et al. [de Geus 2006], who showed that the physiological response to four routes of the same difficulty (7c) varies with the style of the route. Nevertheless, the grades still represent the principal component of the diverse climbing difficulties. The exact meaning of each grade is controversial, as well as their correspondance from countries to countries, but for the non-climbing reader to understand what the grades are, we provide our interpretation of those grades in the table 2.2.

The interpretation for the hardest grade has been established from the comments by Ondra after his world first 9b+ ascent. These comments can be find in [Ondra 2013]: "Success on such a hard route, consists of different factors, such as: huge amount of luck, good conditions, mental strength, and definitely a good shape... and patience!" We have combining this comment with the observation of the climb sequence at time 7'00" of the video (figure 1.1), where the sound expresses the whole difficulty of that sequence.

The design of the in-laboratory bouldering problems has been limited by the size of the wall and the imposed number of holds, that must be the same as the number of available sensors (6). The designed problem grades go from 3 to 5 (Bleau grades), except for one problem which is 7B, designed to show the difference between hard and easy bouldering motions.

Captured data description and post-processing

The effective data capture, after the test sessions for adjusting of camera positions, has been done in eleven half-day sessions, with one climber per session. In average, each session consists of 16 recordings, lasting 30 to 45 seconds. During a recording, 2 to 5 consecutive climbing runs can be done. The hold sensor positions have been changed once in all. The holds mounted on the sensors have been changed at

most once per session, and sometimes between two sessions. The underlying aim of changing the wall configuration was to collect a wide variety of motions. This diversity has been collected to the detriment of the ability to compare performances between climbers.

The main structure of a session consists of one or two warm up recordings, used both for the sporting warm-up of the subject, and for technical adjustments, such as the marker positioning, or the checking the software ability to record. The following six to eight recordings are static motions, where the climber is asking to stay on four holds with a still CoM, or moving the CoM slowly towards a limit position, or to have a rest pose. The other recordings are the explorations by the climbers of all the solutions, i.e. the different climbing motions, he or she can find to the given bouldering problem (1 solution = 1 successful run[5]). For the intermediate or expert climbers on easy problems, this usually leads to about 20 different solutions. For the difficult bouldering problem, only two solutions have been found. The solution exploration took about 10 runs. Therefore, high level climbing requires the adaptation to specific motions. Expert climbers have thus a wider motion repertory than beginners, which can choose their favorite solution among the various set of solution existing for an easy problem.

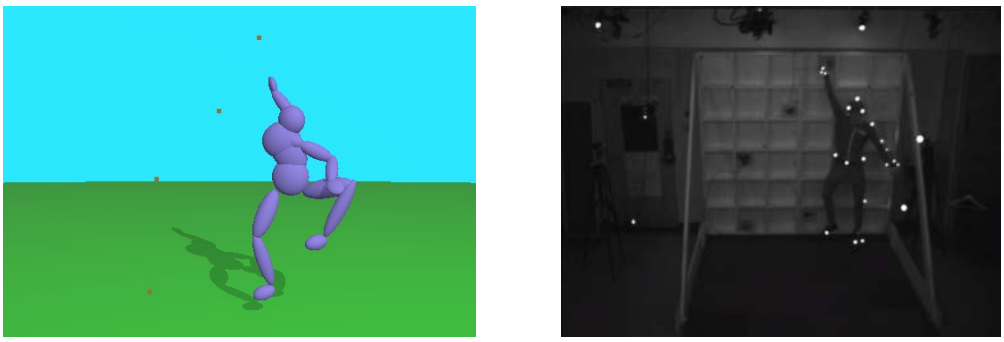
A total of 172 recordings have been captured. In order to the perform the motion analysis presented in the next chapter, a selection has been done to refine these raw data. This selection is presented in the table 2.3. For step 2, the static poses are recordings where the climber is holding on the wall, and applies forces on the sensors without moving. For step 4, the threshold of 100 rad.s^{-1} has been set arbitrarily. It correspond to a joint angle variation of 1 rad (about 60°) between two consecutive frames.

For the step 6, the marker noise is detected. This noise has been noticed in the Arena software, where the marker labeling can be checked. They occured when the labeling of two markers is exchanged repeatedly, on several consecutive frames. This problem has been noticed when some real markers are spatially too close to each other, or when a ghost marker is detected close to a real marker, probably due to a small number of cameras observing that marker. The noise control is done on the linear acceleration of each body segment center. A segment is considered to contain a noisy marker if its acceleration changes sign twice in less than 0.15 seconds.

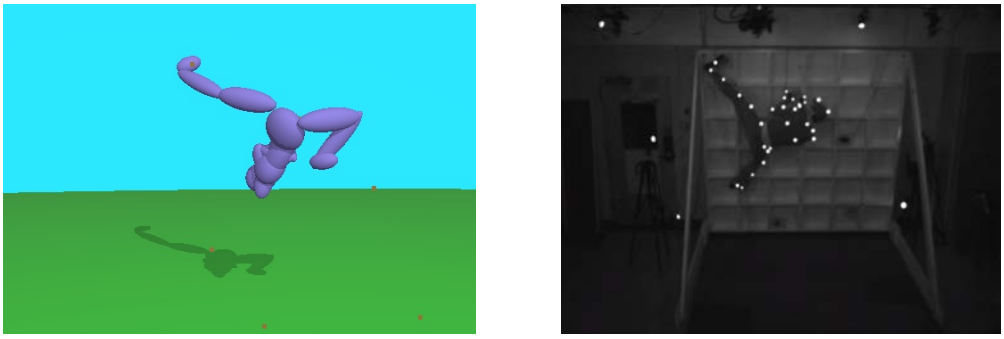
The step 7 is based on two criteria. On one hand, we want to neglect the effect of the joint torque cost at the finger joints, which occures with certain types of holds as describe in section 1.3.1. Consequently, we need motions performed on very good holds. On the other hand, we chose the runs done by the climber who participated to both experiments (the McGill bouldering, and the Top-Rope capture). Those runs are outlined figure 2.17.

In this manuscript, the terms run and sequence are two synonyms. The difference is that a run is a technical climbing term refering to the sequence of moves

[5]A run is a sequence of consecutive moves between when the climber leaves the ground, and when the climb stops, because the last hold is reached or because the climber falls.



(a) *The back markers are hidden by the wall.*



(b) *Marker confusion in a hand matching.*

Figure 2.15: *Two examples of obvious failures (left) in the OptiTrack marker tracking process and their corresponding 2D video images (right). The brown dots on the 3D images are the sensor centers.*

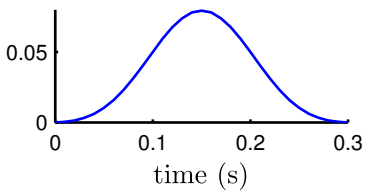


Figure 2.16: *Normalized Blackman covolution kernel of 300ms wide.*

between the instant where the climber leaves the floor and the last hold he reaches before regaining the floor, whereas a sequence is a more generic term. During the capture of the 15 sequences presented in figure 2.17, the wall configuration has not been changed. This configuration is presented figure 2.18. The pictures have been extracted from the grayscale camera closest to the wall.

2.3 Top-Rope 2D-Video Capture

2.3.1 Capture protocol

The 2D-video capture has been done on one afternoon, on a vertical route of the Espace Vertical 2 climbing hall, Grenoble. Three people, including two climbers were present with 3 cameras, two of them being HDV 1080i cameras mounted on tripods. The third one is a mobile camera. The two HDV cameras are positioned on each side of the scene, at 8 meters of each other, and at 6 meters of the wall. They are observing the first 4.5 meters of the route of 11 meters. The mobile camera is between the two others, and follows the climbing ascents by rotating on itself. The three cameras have a resolution of 1920×1080 pixels and a frame rate of 25 Hz (the interleaved frame rate of the HDV 1080i is 50 Hz). The mobile camera is the only one we used currently.

Eight climbing runs have been recorded on the same route, four runs per climber. Seven other video sequences have been taking, one being a rotational scan of the wall from a single point of view, and the others being targeted for the automatic calibration of the fixed cameras. The first climber is an intermediate climber, climbing around 7a (see table 2.2). The second climber is a good climber, having redpointed a 7c+ route. The climbed route is presented in the figure 2.19. This route has been rated at 5a. This low level has been chosen such that we can neglect the effect of the finger muscles.

2.3.2 3D reconstruction of wall

For the 3D reconstruction of the wall, we use the fact that the wall is a vertical plan. From this assuption, we recover the 3D coordinates of the successive hand and foot contacts with the wall. The contact instants are first selected on the video. The images corresponding to contact instants are then extracted from the videos, and the contacts coordinates are computed.

The wall structure is build with regular rectangular planks of 2.44 meters high by 1.22 meters wide. On each image of the moving camera from which the contacts need to be retrieved, the 2D image coordinates of the plank corners are registred. Those 2D points are circled in red on the figure 2.20. Knowing the 3D world coordinates of the point of reference, an homography from the image to the world is computed. The homography computation is done with the fitgoetrans Matlab function. For sequences that do not have enough plank corners, we retrieve other ground truth point coordinates from adjacent calibrated images that are shared by both images.

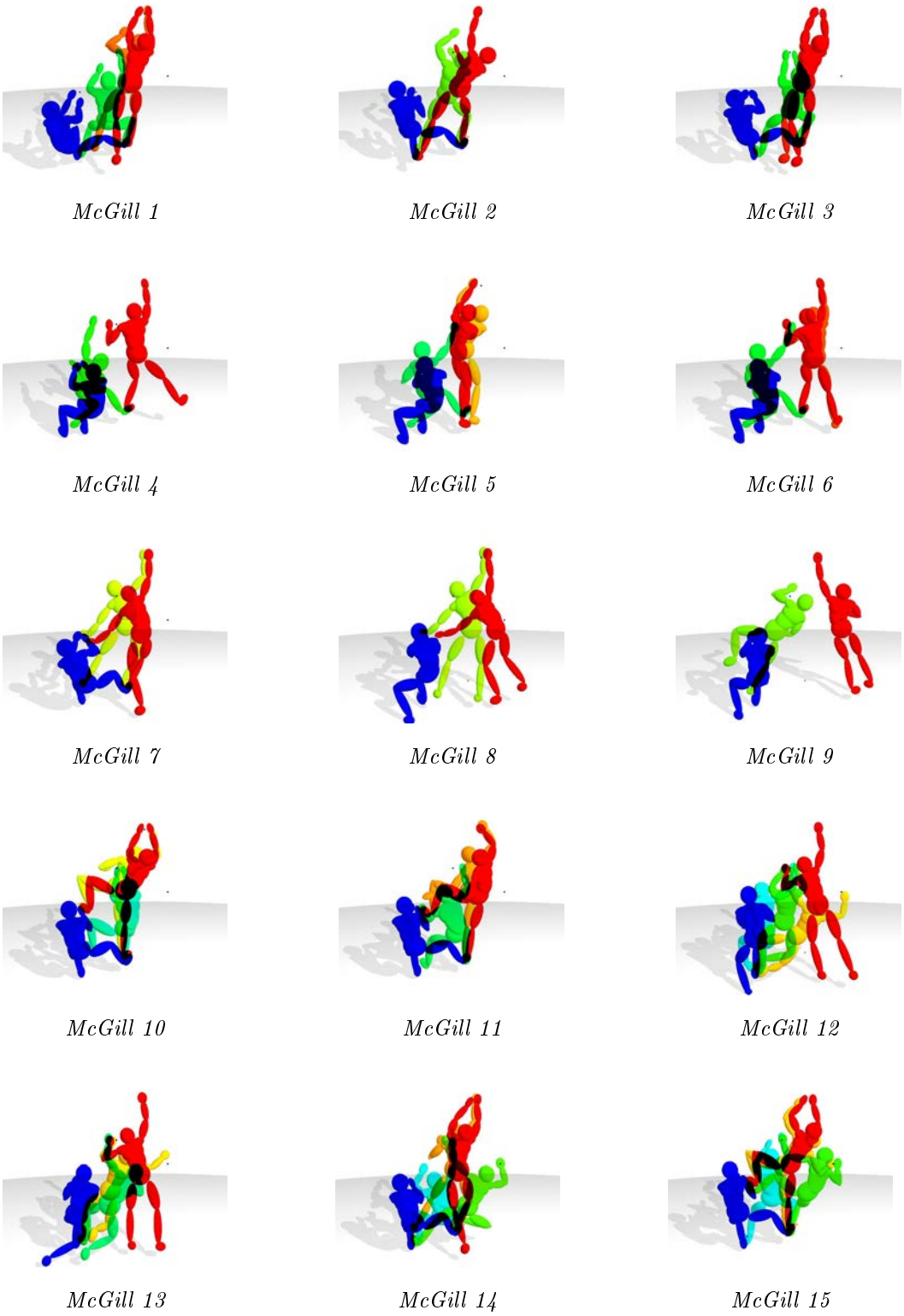


Figure 2.17: *The 15 sequences of our McGill climbing motion database, used for the inverse dynamics evaluation, sampled every 1.5 seconds. The colors correspond to the percentage of time advancement, with the following order: 0% 100%.*

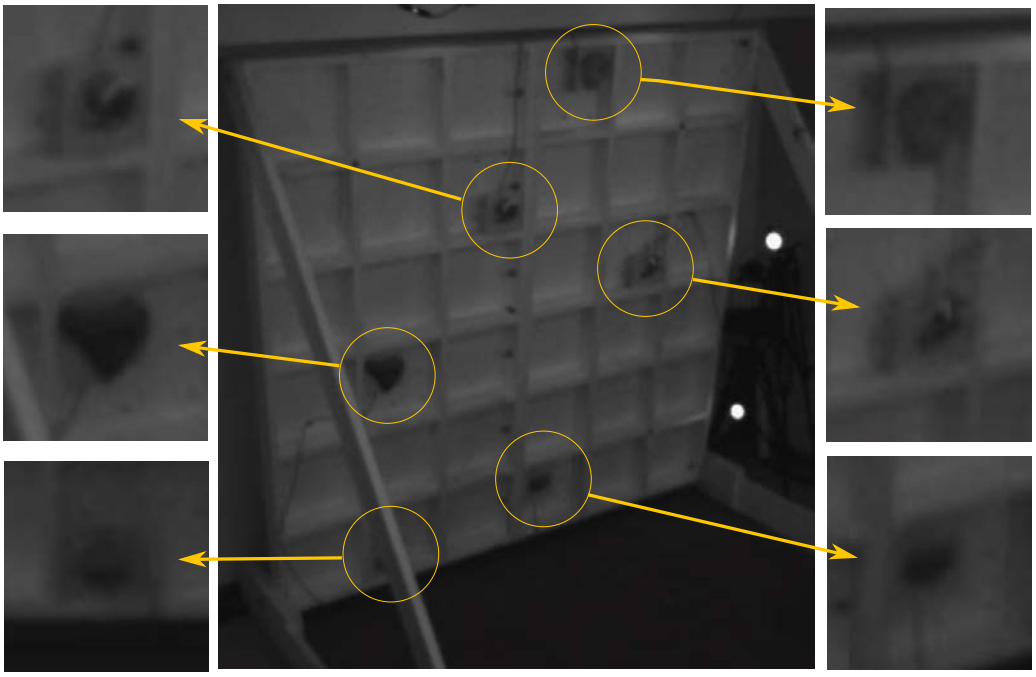


Figure 2.18: *Wall configuration corresponding to the 15 McGill sequences. The close-up of the bottom left hold has been taken from the other grayscale camera, which is farest from the wall than this one. The blur is due to the low camera resolution of* 640×480.

Figure 2.19: *A top-rope climbing sequence captured at Espace Vertical.*

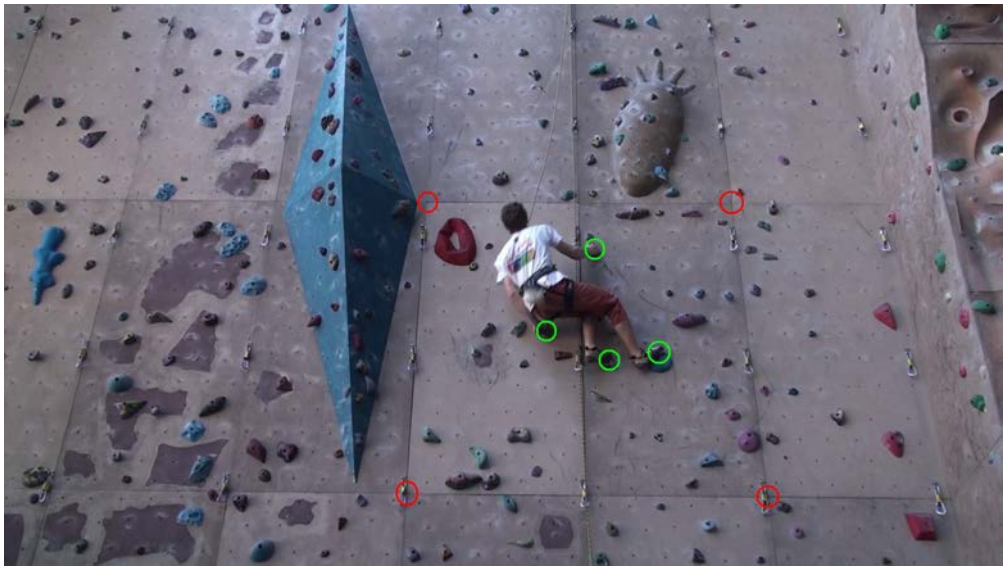
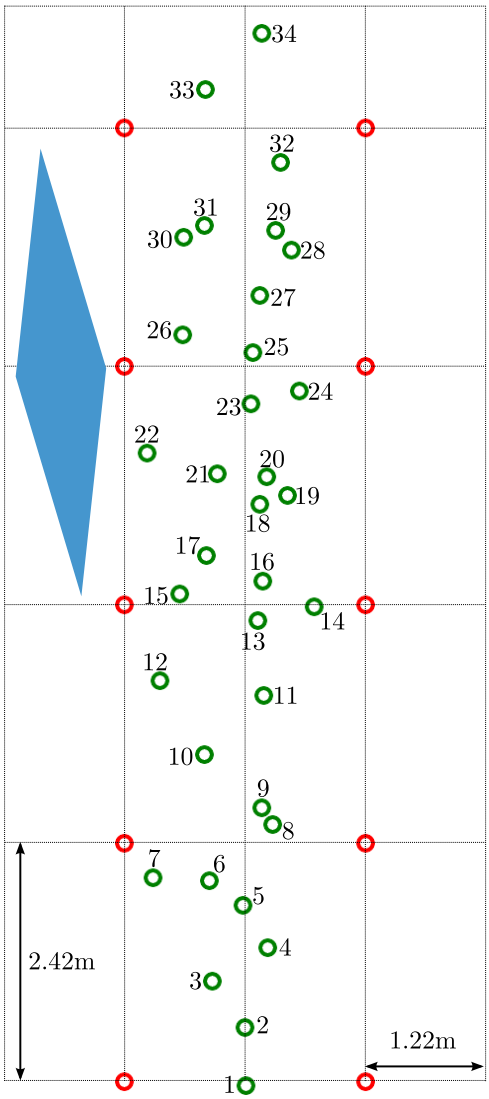


Figure 2.20: *The contact coordinate computation. The wall coordinates of the red points have been measured. From their images coordinates we compute an homography that allows us to retrieve the wall coordinates of the green points from their image coordinates.*

Finally, the 2D contact coordinates are extracted. Those points are drawn with green circles in the figure 2.20. The image coordinates are then converted into 3D world coordinates, by using the computed homographies. The resulting world coordinates are shown in the figure 2.21.

This process has been applied for the cleanest ascent of the climber that also participate to the bouldering capture sessions. During this ascent, 45 contact sets have been identified, leading to a run of 44 movements. We define a movement as the motion between the release of a contact and the establishment of a new contact. There can be either hand or foot movements. A total of 34 contact points have been extracted from those contact sets. Contact points corresponding to the same hold on several different contact sets have been merged by computing their average world coordinates. Note that with 44 movements, the climb is necessarily aerobic. In other words, the climber needs to breath in order to climb, which is usually not the case for bouldering movements.

If the wall would not have been planar, the 3D reconstruction could have been done by using a structure from motion approach. As a proof of concept, we reconstructed one of the bouldering walls of the University of Grenoble, as shown in figure 2.22. This reconstruction has been done using the system VisualSFM by Wu [Wu 2011]. The input pictures have been taken with a Canon EOS 7D, of resolution 5184×3456 pixels. We took 45 different points of view, at about 3 and 5 meters of the wall, with heights between 0.5 and 2.5 meters.



(a) *The numbering of the route holds.*

pose num	set of holds for each pose			
	foot		hand	
	right	left	right	left
1	1	2	5	6
2	3	2	5	6
3	3	2	9	6
4	4	2	9	6
5	4	5	9	6
6	4	5	9	10
7	6	5	9	10
8	6	7	9	10
9	6	7	11	10
10	6	6	11	10
11	9	6	11	10
⋮	⋮	⋮	⋮	⋮
35	24	23	27	26
36	24	23	27	31
37	25	23	27	31
38	25	26	27	31
39	25	26	32	31
40	25	27	32	31
41	28	27	32	31
42	28	27	32	33
43	29	27	32	33
44	29	30	32	33
45	29	30	34	33

(b) *The ordered sequence of contacts, as observed in the video. The pose num column contains pose numbers. The other columns contain hold numbers.*

Figure 2.21: *The 2D contact coordinates after homography computation (a), and the sequence of contacts (b), observed in the sequence of figure 2.19. The red circles are used to compute the image-to-world transformation. The green circles are the retrieved contact points.*

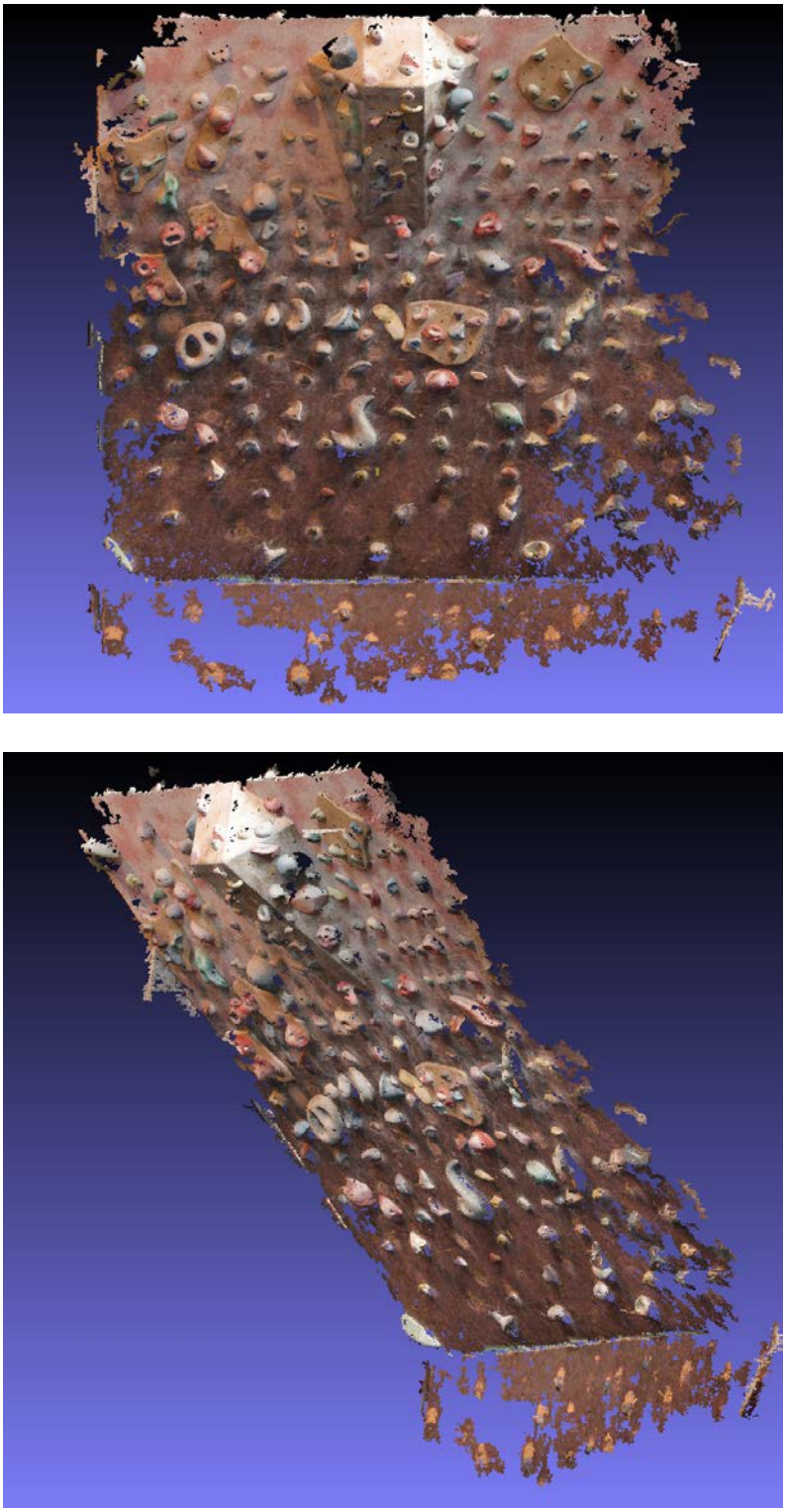


Figure 2.22: *Proof of concept of the 3D reconstruction of a non-planar wall, using VisualSFM [Wu 2011]. The two images are the front view and the side view of a textured point cloud (2 million points), reconstructed from 45 high-definition images. (see text for details)*

Chapter 3

Inverse Dynamics with Contact Force Prediction

Contents

3.1 Introduction **49**
3.2 Related Work **51**
3.3 Inverse Dynamics Formulation **52**
3.3.1 Equation of motion 53
3.3.2 Disambiguation by minimization of torques and forces 54
3.3.3 Contact transition model 56
3.3.4 Joint torque weighting 58
3.4 Experiments and Validation **58**
3.4.1 Standard inverse dynamics examples 60
3.4.2 Climbing with limb specific weights 61
3.4.3 Climbing with anatomical weights 62
3.5 Conclusion **71**

3.1 Introduction

In this chapter, we present a method to estimate contact forces and joint torques of a human performance from motion data only. This is known to be an ill-posed problem as several force configurations are equally consistent with the laws of physics for a single given kinematic configuration. Along the line of previous works in the estimation of the dynamics of motion, we improve the accuracy of the estimation of the exact values of contact forces. Our approach is based on minimization of joint torques and considers contact forces as free variables. This optimization introduces weighting parameters which are used to enforce smooth transition of contacts and muscle preferences of the performer while respecting the laws of physics as a hard constraint. We have tested two categories of motion, walking at different speed as standard inverse dynamics examples and climbing on an artificial wall as it typically yields challenging situations with multiple contacts. We demonstrate that our framework can estimate contact forces up to 5% of the body weight (%BW) for walking and 22%BW for climbing on the McGill dataset, by comparison with ground truth measurement from force plates and sensors.

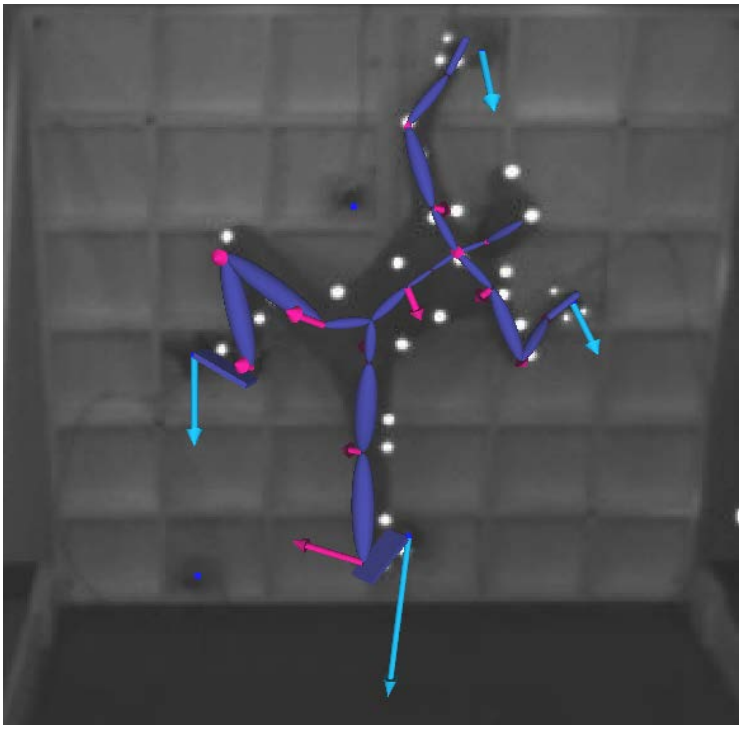


Figure 3.1: *Prediction of contact forces and joint torques. Blue arrows correspond to contact forces, magenta arrows to torque axis, magnitude is proportional to amplitude.*

Contacts shape the interaction with the environment. They support locomotion through their cooperation with the internal torques applied by muscles. For a given mechanical model of articulated bodies, once contact forces are known, internal torques at joints can be computed from kinematic data. This process is known as Inverse Dynamics. In biomechanics, the method of choice for measuring contact forces is to use force plates. While extremely accurate, they are costly and cumbersome to operate. For motion analysis, there is a clear interest to be able to evaluate contact forces, and thus internal torques, from motion data only, removing the needed of force sensors. This motion data can be obtained either from motion capture or from video tracking. The pioneering work of Brubaker et al. [Brubaker 2009] showed that location and timing of contacts could be automatically inferred from motion data. Their work relies on a parametric model of contact which approximates the true forces. We build on this approach as a first step for contact detection but with the goal of delivering accurate measurement of the true contact forces from motion data only, without a parametric model. Besides the application in biomechanics for motion analysis, Inverse Dynamics is also a useful component for physically based motion synthesis approaches: given a kinematic plan, Inverse Dynamics provides the necessary torques to apply. As such, estimation of dynamics from video also benefits to physically-based animation techniques [Vondrak 2012b].

Estimation of multicontact forces from kinematic data alone is fundamentally an ill-posed problem. Typically, from the observation of only the body pose of a person hanging on a bar with both hands, nothing reveals if the person is using strong forces and torques on the left arm only, or the reverse, or, most likely, a balance between left and right arms. This ambiguity is even more critical for complex contact scenarios such as climbing, which are addressed in this thesis, and have constituted the primary motivation for the development of our method. The overall approach of this chapter is to evaluate how well this ambiguity can be resolved using a criteria of minimal internal torques. Our formulation does not rely on a parametric model of contact and delivers an accurate estimation of the forces. Recent work on inverse dynamics from the robotic literature for motion control of autonomous robots follows a similar strategy [Righetti 2013], as well as recent work in biomechanics [Robert 2013]. The core of our method is to characterize the ambiguity of the multiple contact points through the choice of a set of weighting parameters in an optimization for minimal internal torques and accurate estimation of the values of contact forces across time.

We show for locomotion and climbing tasks that default values for weighting parameters are already enough to produce accurate estimates of contact forces. All of our motion data have been recorded in parallel with exact measurement of contact forces for ground truth evaluation. We show results for locomotion and climbing on an artificial wall as a situation with complex interactions and multiple contacts.

3.2 Related Work

Extraction of physical properties from video has been explored for free fall of rigid bodies [Bhat 2002], and more recently with instantaneous impacts [Salzmann 2011]. We focus here on a specific problem related to human motion. Similar to that of Brubaker et al. [Brubaker 2009], the purpose of our work is to provide methods to estimate contact forces directly from motion data only, obtained by motion capture or by video tracking [Balan 2005, Urtasun 2006]. We show that a concise formulation allows us to evaluate the exact contact force value without the need for a parametric model. The work of Brubaker et al. [Brubaker 2009] estimates both the location and timing of contacts. Our work is a continuation in this line of thought in an attempt to show that vision-based system can be used to deliver measurements of dynamics. The parametric model of Brubaker et al. [Brubaker 2009] allows the detection of contacts, but it does not always agree with accurate measurement of contact forces. Typically, their parametric model generates contact forces with respect to the local geometrical configuration at the point of contact only. In contrast, our formulation takes into account the global configuration of body posture and its impact on contact forces, even if the local geometry of the contact point is constant. Using a criteria of joint torque minimum, we show through evaluation with ground truth data from force plates that we can improve the accuracy of contact force estimates.

The criterion of optimality has been recently explored in robotics by Righetti et

al. [Righetti 2013]. Inverse dynamics is used in their case to control an autonomous robot. No contact forces are measured to plan the torque motor program, and these forces must be evaluated automatically. We share the same requirement but with a goal related to motion analysis and biomechanics. In particular, we show that accurate estimation of contact forces is possible for human locomotion and we compare to ground truth. Interestingly, such an optimization-based approach introduces parameters that can be related to the strength of muscles. The importance of the setting of those parameters has been studied by Robert et al. [Robert 2013]. Their experience consists in standing up from the seat of a car with a helping handle for the hand, which is a multicontact scenario. Some contact forces are measured and used in the contact force estimation, some forces are measured and used as ground truth and the other contact forces are not measured and are recovered by the algorithm. They showed that the ground truth forces are better recovered with joint torque weights proportional to the joint torque limits than with uniform joint torque weights. We find the converse result in our case.

Our work is also related to previous attempts in the computer animation community to estimate physical parameters from motion capture in order to re-used motion in a physical simulation [Liu 2005]. This work combines a human body model with a penalty-based model for contacts and rest pose, which introduces stiffness parameters that must be estimated. We formulate the parameter estimation problem in a more general way and show that the true contact forces can be recovered. Our formulation introduces weights to the optimization that relate to the physiological strength of muscles in a more general way, without the bias of a parametric model.

Several recent works show the benefit of using physics modeling as a prior for human motion tracking [Vondrak 2012a], and for human motion recovery [Wei 2010, Vondrak 2012b]. It allows the integration of additional information to avoid foot skating, predicts a few time steps (frames) in advance or brings additional motion constraints to monocular videos based on dynamic information. At the core of physical modeling is the definition of a good contact model, and we propose here a method to estimate these forces reliably.

Contacts are continuously established and released across time. For motion synthesis, Mordatch et al. [Mordatch 2012] present a similar optimization-based framework to ours, which evaluates possible contacts and intensities. In their case, they split motion into phases with discrete changes of weights in the optimization of contact forces. We introduce a parametric model for contact transition which accounts for the natural timing of interaction with the environment.

3.3 Inverse Dynamics Formulation

Our input for the inverse dynamics problem consists solely of kinematic data. The output consists of the contact forces and the joint torques to actuate body links. This is an ill-posed problem and our approach is to optimize for minimal torques. Contact forces are also introduced into the optimization through a parametric model

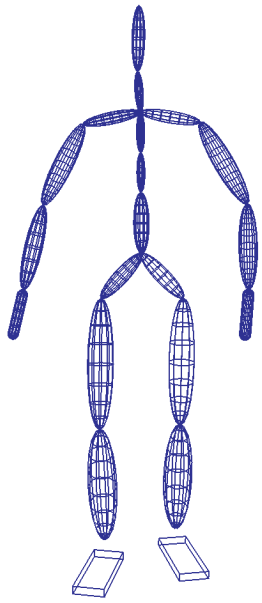


Figure 3.2: *Articulated model of rigid bodies*

of contact transitions.

3.3.1 Equation of motion

The pose, or configuration, of the body is parameterized with a set of $n+6$ degrees of freedom (DOF), with $n = 43$ for the joints and 6 DOFs for the root body (Figure 3.2). Every DOF is actuated by muscle torques, except the 6 DOF of the root body which are not actuated. The equations of motion are thus split between actuated DOFs and non-actuated DOFs. Conceptually, the root position and orientation are actuated through the contact forces with the environment, but up to the ambiguity of multiple joint torque activations respecting the contact constraints and the enforcement of physical laws. Rotations are parameterized using exponential maps to avoid Euler angles issues (gimbal lock, double coverage) or the normalization issues of unitary quaternions [Murray 1994].

We formulate the equation of motion using generalized coordinates:

$$\mathbf{M}\,\ddot{\mathbf{q}} + \mathbf{h} = \mathbf{S}^\top \boldsymbol{\tau} + \mathbf{J}_\mathrm{c}^\top \boldsymbol{\lambda} \tag{3.1}$$

where $\mathbf{M} \in \mathbb{R}^{(n+6)\times(n+6)}$ is the generalized mass matrix, $\mathbf{q} \in \mathbb{R}^{n+6}$ are the generalized coordinates, $\mathbf{h} \in \mathbb{R}^{n+6}$ accounts for the Coriolis, centrifugal and gravity effects, $\mathbf{S} = [\mathbf{I}_{n\times n} \quad 0_{n\times 6}] \in \mathbb{R}^{n\times(n+6)}$ defines the joint-selection matrix, $\boldsymbol{\tau} \in \mathbb{R}^n$ are the torque for actuation of non-root joints, $\boldsymbol{\lambda} \in \mathbb{R}^k$ are the contact forces, and $\mathbf{J}_\mathrm{c} \in \mathbb{R}^{k\times(n+6)}$ is the contact Jacobian, mapping contact forces from contact point to generalized coordinates.

We split this equation between the actuated joint part

$$\boldsymbol{\tau} = \mathbf{S}\left(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{h} - \mathbf{J}_c^\top \boldsymbol{\lambda}\right) \tag{3.2}$$

and the non-actuated root part

$$\overline{\mathbf{S}}\left(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{h}\right) = \overline{\mathbf{S}}\ \mathbf{J}_c^\top \boldsymbol{\lambda} \tag{3.3}$$

where $\overline{\mathbf{S}} = [0_{6\times n}\ \ \mathbf{I}_{6\times 6}] \in \mathbb{R}^{6\times(n+6)}$ is the root-selection matrix.

If there are more than two contact points ($k > 6$), this last equation cannot be used to recover the contact forces $\boldsymbol{\lambda}$ from kinematic data only. There is an infinite number of solutions which respect the constraints and laws of physics. Similar to the work of Righetti et al. [Righetti 2013], we follow a criteria of optimality on contact forces and internal torques. In their case, the goal is the autonomous actuation of the robot, so they focus on the derivation of a torque controller. In our case, we focus on motion analysis and the estimation of contact forces. In addition, we introduce a specific model for continuous contact transitions, to deal with the appearance and disappearance of contacts.

3.3.2 Disambiguation by minimization of torques and forces

The minimization of external forces is formulated using two quadratic functions for the torques and the contact forces. That is,

$$\min_{\boldsymbol{\tau},\boldsymbol{\lambda}}\ \frac{1}{2}\boldsymbol{\tau}^\top \mathbf{W}_\tau \boldsymbol{\tau} + \frac{1}{2}\boldsymbol{\lambda}^\top \mathbf{W}_\lambda \boldsymbol{\lambda} \tag{3.4}$$

where $\mathbf{W}_\tau \in \mathbb{R}^{n\times n}$, and $\mathbf{W}_\lambda \in \mathbb{R}^{k\times k}$ are definite positive diagonal matrices.

To reduce the number of unknowns and the number of constraint equations, we implicitly include the actuation part of the Newton-Euler equation in the minimization, as in [Wei 2010], by injecting equation 3.2 into equation 3.4. This gives

$$\min_{\boldsymbol{\lambda}}\ \frac{1}{2}\boldsymbol{\lambda}^\top\left(\mathbf{J}_c\mathbf{S}^\top\mathbf{W}_\tau\mathbf{S}\mathbf{J}_c^\top + \mathbf{W}_\lambda\right)\boldsymbol{\lambda}\ -\ (\mathbf{M}\ddot{\mathbf{q}} + \mathbf{h})^\top\mathbf{S}^\top\mathbf{W}_\tau\mathbf{S}\mathbf{J}_c^\top\boldsymbol{\lambda} \tag{3.5}$$

such that

$$\overline{\mathbf{S}}\ \mathbf{J}_c^\top\boldsymbol{\lambda} = \overline{\mathbf{S}}\left(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{h}\right) \tag{3.6}$$

$$\mathbf{F}\boldsymbol{\lambda} \leq 0 \tag{3.7}$$

where $\mathbf{F} \in \mathbb{R}^{d\times k}$ is polyhedral friction constraint matrix. We use a Coulomb model to account for frictional effect of contacts [Stewart 1996]. $\mathbf{F}$ projects the contact forces on the outward normals of the facets of the polyhedral friction cone. We do not use the classical multiple force components on the cone generative vectors, as this would lead to much more unknowns.

To solve the system defined by equations 3.5, 3.6, and 3.7, we must define the positive definite diagonal matrices $\mathbf{W}_\tau$ and $\mathbf{W}_\lambda$. The values of these coefficients directly impact the result of the inverse dynamics computation. As a first intuition, when a contact is established, the cost of the contact force must be much lower

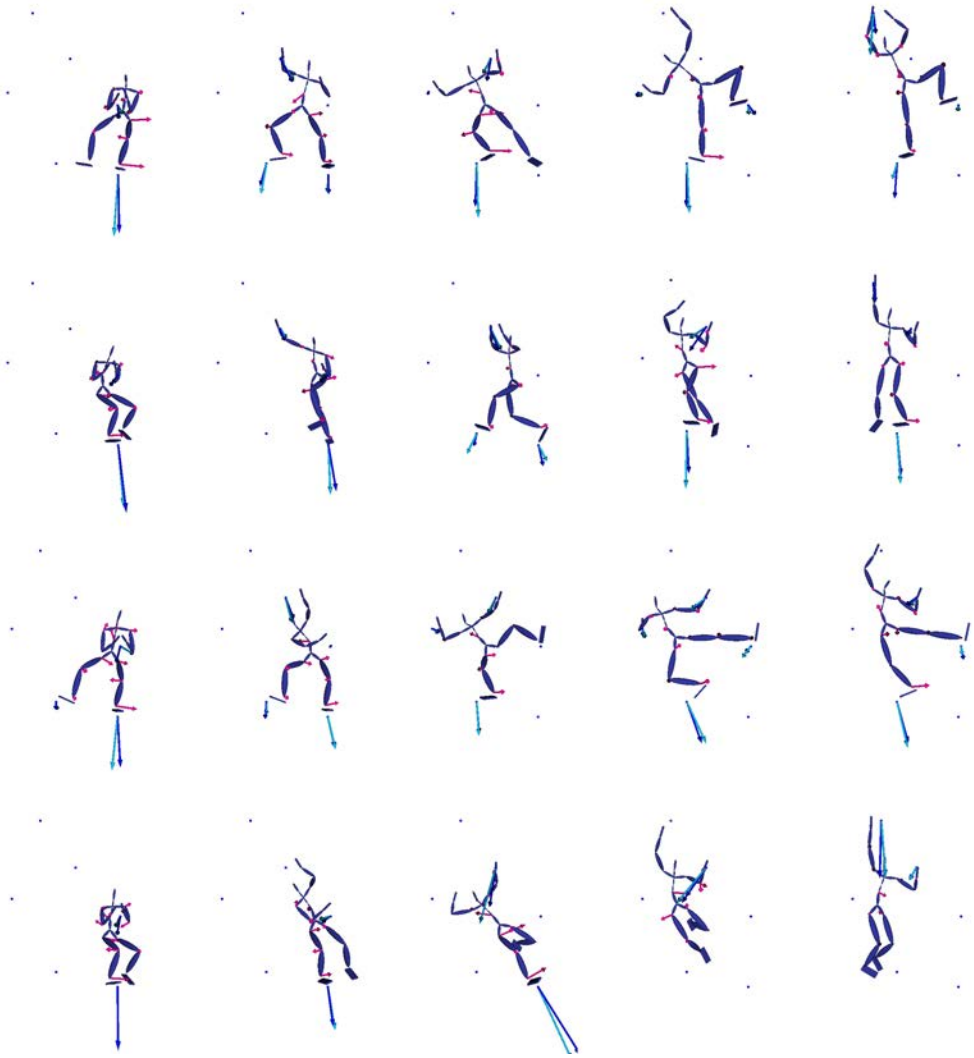


Figure 3.3: *Prediction of contact forces (Dark blue, ground truth - Light blue, prediction) and joint torques (magenta) for climbing.*

than the cost of the joint torques, such that the joint torques rule the contact force generation. Conversely, this contact cost must be much higher than the joint torque cost when the contact is broken, in order to obtained the joint torques associated to none in-air contact forces. Consequently, we affect a high value to coefficients of $\mathbf{W}_\lambda$ when no contact occurs, and a low value when a contact is established. Our experiments have shown that the values 10^4 and 10^{-4}, respectively, provide good results. It logically follows that $\mathbf{W}_\tau$ needs to be set to an intermediate value between these two extremes. In order to account for the fact that legs produce more torques than arms, we set in a first approximation a weight value of 3000 to arms, 300 for the knees and 100 for all other actuators.

Creation and localization of the contact points are geometrically determined. The approach of Brubaker et al. [Brubaker 2009] uses a penalty-based approach with a sigmoid on the distance of the contact to the ground, and could be used here as preprocess to estimate contact locations and timing. For the sake of simplifying the implementation, we perform a geometric detection of contact locations and use a threshold on the velocity of some reference points to establish contacts at the hands and feet. For noisy input motions, an external user adjusts the contact instants thanks to a graphics user interface. If the contact is established, then its $\mathbf{W}_\lambda$ is set to the low value in order to only weakly penalize the contact forces; otherwise, it is set to a high value. In human locomotion, contacts are not reduced to impulses, but instead follow a continuous trajectory, increasing and decreasing smoothly. We show now how we model this property.

3.3.3 Contact transition model

On contact detection, a first approach would be to switch the value of the contact weights $\mathbf{W}_\lambda$ directly from the high to the low value in a single discretized time step. This results in discontinuities on contact transitions, both in the estimation of contact forces and internal torques. Instead, we use a parametric model of the transition. The transition is modeled as a linear interpolation of an exponent over a time duration of $250ms$, between the high value of 10^4 for a transition from no contact to 10^{-4} for full contact:

$$w(\alpha_t) = 10^{-4\alpha_t + 4(1-\alpha_t)} \tag{3.8}$$

where $\alpha_t \in [0, 1]$ parameterizes the timing between the two contact events over a $250ms$ window, from no contact to full contact. The equation is easily inverted for the reverse phenomenon.

The values of this parametric model, amplitude and timing, have been determined experimentally and prove to work well for our case. As future work, we plan to optimize them directly from the data using a parametric formulation. Such an approach of continuous weights in optimization of contact forces can be related to the recent works of Mordatch et al. [Mordatch 2012] for motion synthesis where contact locations are optimized to fulfill a kinematic objective.

Joint	Motion	Robert et al.	McGill
Hip	Flexion	185	89
	Extension	190	157
	Abduction	190	243
	Adduction	190	86
	Internal Rotation	60	90
	External Rotation	60	38
Knee	Flexion	100	110
	Extension	168	142
	Internal Rotation	20	44
	External Rotation	20	124
Ankle	Dorsiflexion	126	12
	Plantarflexion	126	164
	Inversion	20	45
	Eversion	20	105
Shoulder	Flexion	92	64
	Extension	67	175
	Abduction	71	50
	Adduction	67	145
	Internal Rotation	52	154
	External Rotation	33	19
Elbow	Flexion	77	112
	Extension	46	78
	Pronation	15	21
	Supination	15	77
Wrist	Flexion	185	102
	Extension	190	31
	Abduction	190	73
	Adduction	190	52
Lumbar	Flexion	143	159
	Extension	234	84
	Right Bending	159	115
	Left Bending	159	122
	Left Axial Rotation	∅	95
	Right Axial Rotation	∅	91
Neck	Flexion	100	7
	Extension	100	2
	Right Bending	∅	3
	Left Bending	∅	5
	Left Axial Rotation	∅	1
	Right Axial Rotation	∅	1

Table 3.1: *Joint torque limits (N.m) obtained from the literature [Robert 2013] and from the McGill dataset.*

In order to be useful for animation or biomechanics, continuity in estimation of contact forces and joint torques is an important issue. As a first approach, one might consider that results from the naive transition model could be smoothed. Doing so would result in modifying the value of forces in a way which is violating the laws of physics. By smoothing the weight parameters of the optimization rather than the output of this optimization, it allows to perform a filtering of the forces while staying in the null-space of the under-constrained structure of the inverse dynamics problem, and as such laws of physics are strictly respected.

3.3.4 Joint torque weighting

Four different joint torque weightings have been evaluated. Each of them are normalized at 10^3 such that the weights lie in the upper part of the range of the contact force weights $\left[10^{-4}, 10^4\right]$. The lower part of this range is reserved for the crossing of the contact weights, occurring for instance during the double stance phase of walking motions. The two first weightings operate on the projection of the joint torque on the basis vectors of the joint frames of the character. The first one contains uniform weights. The second one, the limb specific weighting, has weights of 3000 for the arms, 300 for the knees and 100 for all other actuators. We call the two others the quadratic and the linear weights. They both operate on the projection of the joint torques on anatomical axes. The definition of these axes follows the recommendations of the International Society of Biomechanics [Wu 1995, Wu 2002, Wu 2005]. To visualize them, we used the software OpenSim [Delp 2007]. The quadratic weights is the squared inverse of standard joint torque limits, as in [Robert 2013]. The linear weights comes from our specific measurement of joint torque limits from the McGill climbing database (figure 2.17). These limits are the maximal torques obtained with sensor based inverse dynamics, i.e. by transferring the contact forces measured by the sensors into joint torques (equation 3.2). The weights are then the absolute inverse of these torque limits. The table 3.1 contains the two sets of limits we used.

3.4 Experiments and Validation

We explore two categories of motion: walking and climbing. For both, we compare our results with ground truth data. Input kinematic data are obtained from motion capture.

For the walking sequences, we first used data from OpenSim software [Delp 2007]. OpenSim is widely used in the biomechanics community and we used the data it provides for a walking example as ground truth for both contact forces and joint torques. We performed additional experiments with a standard force plate for a walk at different speed.

Climbing on artificial wall presents interesting challenge as it typically involves multiple contact points. In order to compare to ground truth data in this situation, we equipped an artificial climbing wall with force sensors on six holds for about a 2.5

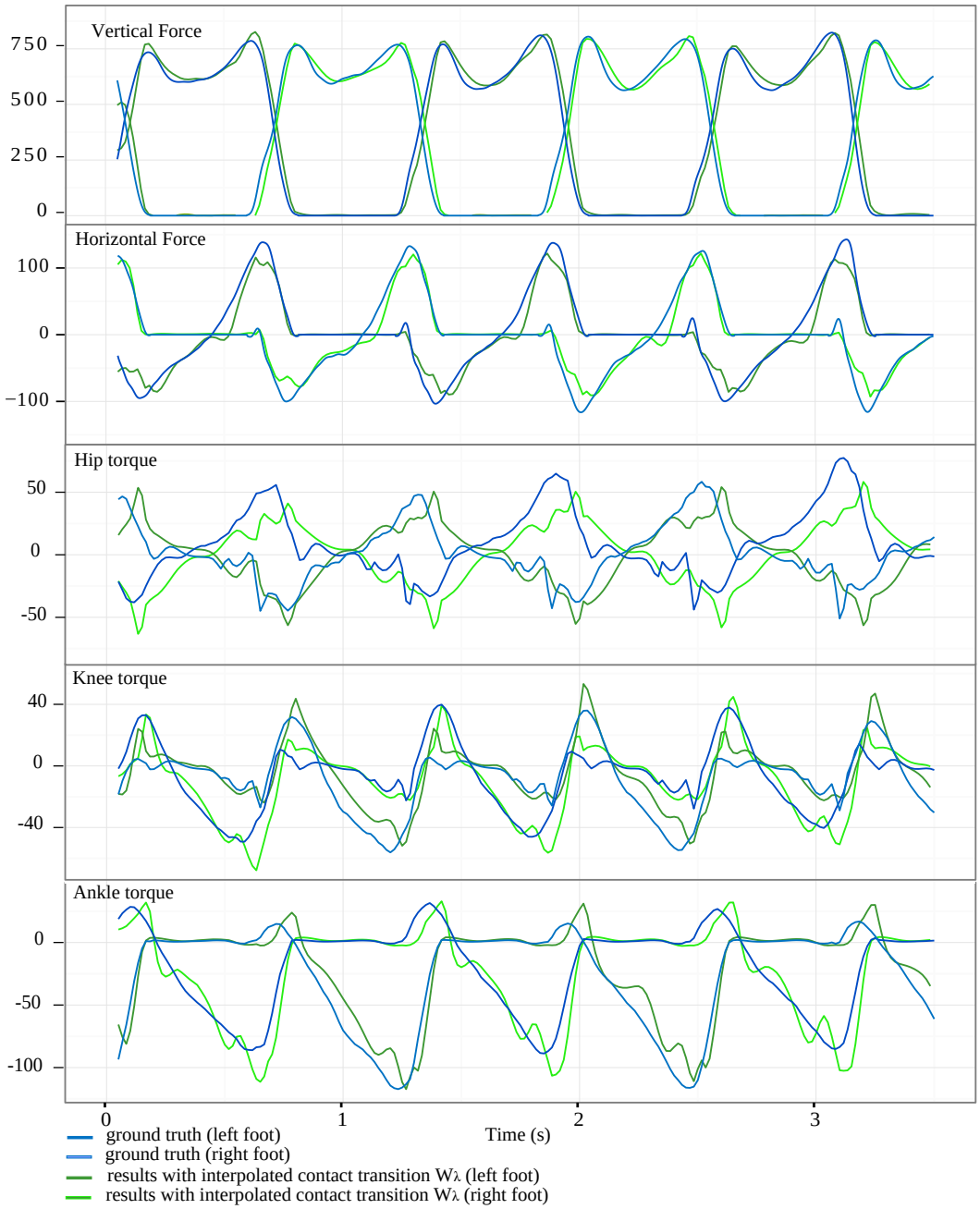


Figure 3.4: *Prediction of contact forces at both feet and joint torques at hip, knee and ankle for walking.*

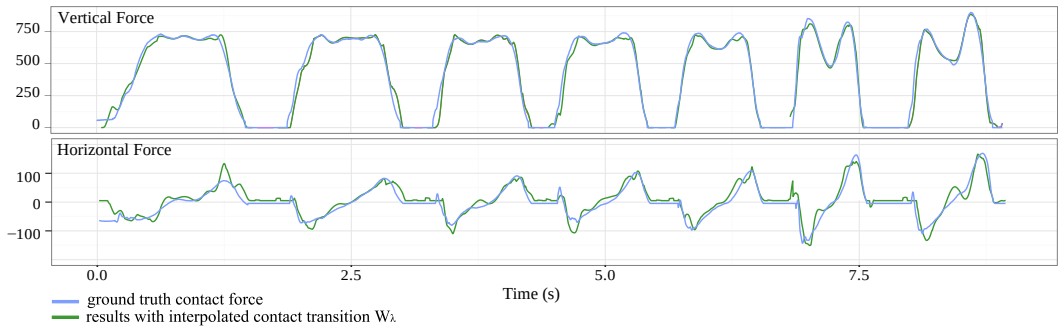


Figure 3.5: *Prediction of contact forces at one feet for walking at increasing speed.*

meters vertical range. We evaluate here the quality of the prediction of the contact forces.

3.4.1 Standard inverse dynamics examples

Reference walking As a first evaluation, we used the kinematic and dynamic data proposed for the walking example in the OpenSim software. Figure 3.4 shows the comparison between our estimation of contact forces and joint torques with contact forces measurement and the computed joint torques from OpenSim. The contact establishments and breakings are modeled with a transitions window of $250ms$. The vertical and horizontal contact forces are quite well recovered. The average shapes of the joint torques are also well computed, but the detailed shapes differ. We noticed experimentally that the shape of the feet influences a lot the joint torques, whereas the contact forces were less sensitive to this parameter. The shape of each foot is represented by three points of contact with the ground. The location of these points is procedurally deduced from a geometrical model of the foot. As the foot is rotating during the stance phase, these points transit from the heel to the toe, bounding implicitly the position of the foot center of pressure.

Walking at different speed We performed an experiment of walking with increasing velocity. As the subject walks faster, it is interesting to observe how the profile of contacts evolves. As shown by the force plate measurement, the slow walk exhibits a rather flat profile during contact. As the speed increases, the contact evolves toward a situation with two maxima, one at the reception of the foot at heel strike, and one at toe off as the contact is used to push the foot forward. In addition, some forces increase progressively on the axis parallel to the walking direction. Figure 3.5 shows the comparison between the measured ground reaction force and our prediction. It shows that our approach is able to recover the precise profile of contact forces.

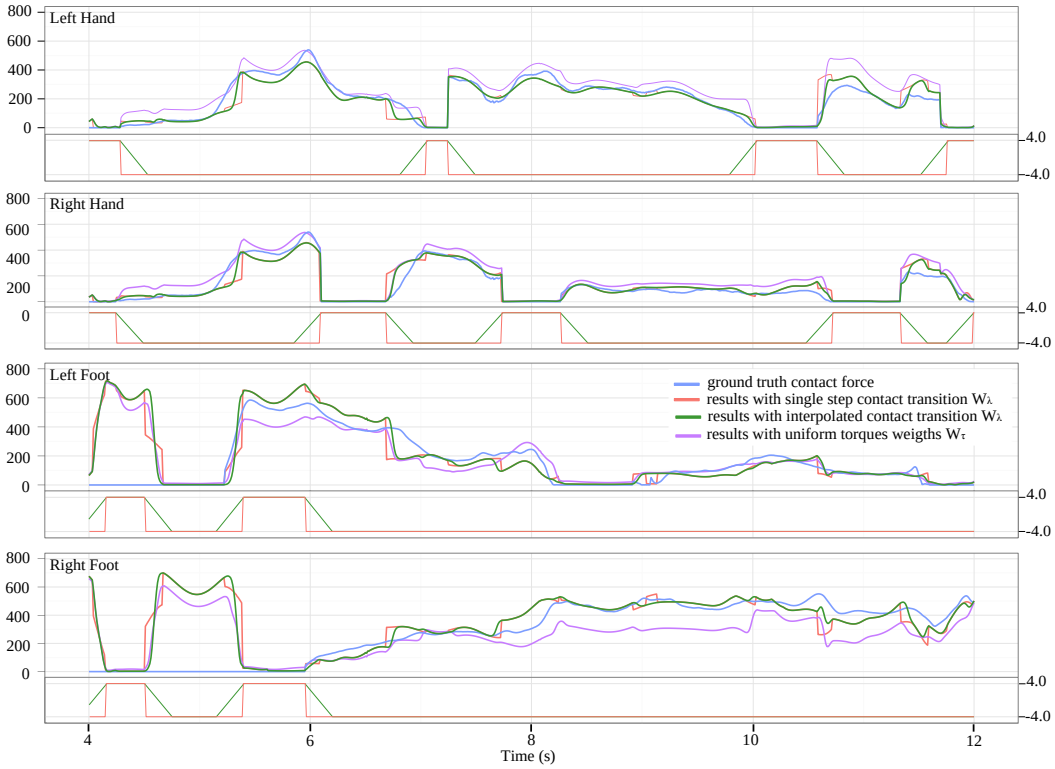


Figure 3.6: *Prediction of contact forces at hands and feet for climbing. For each of the four contacts, the subplot at the bottom shows the contact transition, single step (orange) and interpolated (green). The other subplot shows contact forces from the ground truth data (blue), single step contact transition with limb specific weights (orange), interpolated contact transition with limb specific weights (green), and interpolated contact transition with uniform weights (purple).*

3.4.2 Climbing with limb specific weights

Climbing inherently involves several simultaneous contact points at hands and feet. As such, it presents an interesting challenge for the ill-posed problem of contact estimation from kinematic data only. Similar to our walking validation, we compare the climbing results of our approach to ground truth data. Figure 3.6 details the results for four contacts (hands and feet). For each contact, we use a single point of application for the force, geometrically located at the body part in contact.

Figure 3.6 illustrates how the transition model for a given contact improves not only the estimation of the associated force, but also the entire system. It clearly shows the benefit of having a global approach for contact modeling rather than a local parametric model for the problem of force value estimation. In addition, the figure 3.6 compares the effect of specializing torque distribution between joints using the $\mathbf{W}_\tau$ matrix. Compared to a uniform distribution, affecting more weights to legs by decreasing the specific value in $\mathbf{W}_\tau$ improves the results. At the beginning of the sequence, the subject is approaching the wall by walking. This explains why contact

forces are estimated at the feet but do not appear on the ground truth measurement. Note that these forces follow a standard walking pattern at the beginning. Figure 3.3 shows 3D results for a selection of frames.

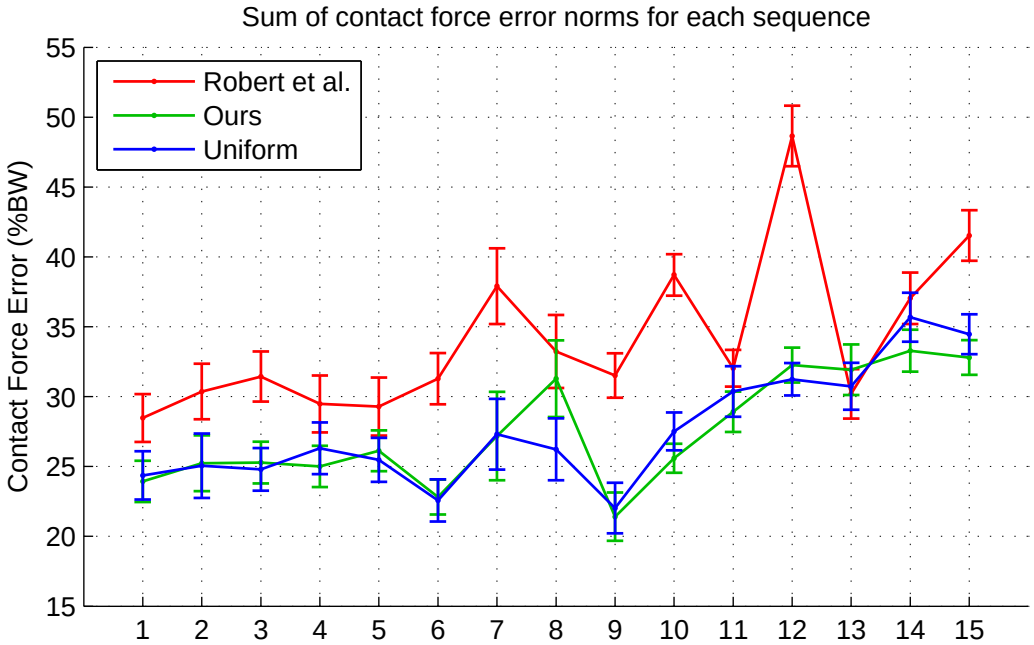
3.4.3 Climbing with anatomical weights

State-of-the-art evaluation The 15 sequences of the McGill database are used to evaluate the approach of Robert et al. [Robert 2013], which uses the quadratic weights defined previously. We compare them with the uniform weights. The red and blue curves of the figure 3.7a show the 95% confidence intervals of the sum of the norms of the 4 differences between the estimated contact forces by inverse dynamics, and the measured contact forces. The uniform weights give significantly less errors in 11 cases over 15, and with 68% of confidence, the error is decreased in all cases. The weighting of [Robert 2013] is thus less suited than a uniform weighting for our climbing database. But in [Robert 2013], quadratic weights gave better results than uniform weights, therefore, the ideal weighting might depends on the observed activity.

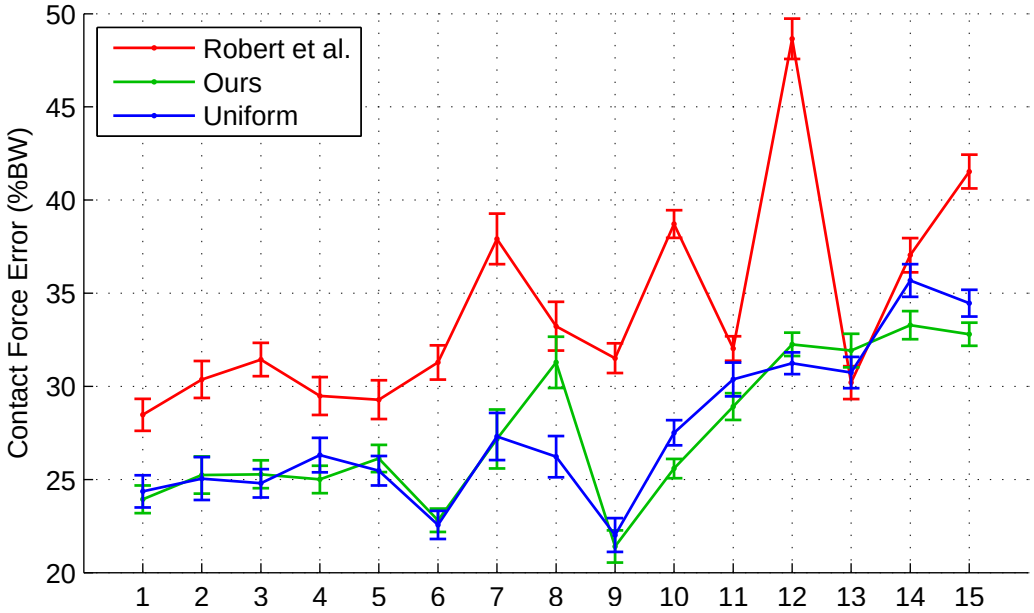
Activity-specific joint torque limits The linear weights learned on the McGill dataset are used to obtain the green curve in the figure 3.7. In 12 cases out of 15, our torque weights decrease significantly (95%) the contact force errors, when compared with the weighting of Robert et al. [Robert 2013], and has no significant effect on the other cases. The comparison between our weighting versus a uniform weighting at 95% confidence leads to no significance differences except in one case (McGill 8) where the uniform weights are better. With 68% confidence, our weighting is significantly better in 3 cases out of 15, worse for the sequence 8 and not significantly different for the other cases. The sequence 8 is plotted at 4 fps in figure 3.8. The main move of this climbing run is a dyno (i.e. a jump) without the left foot. This high speed motion make it difficult to capture. To quantify this difficulty, we call the inertia error the quantity

$$I_{err} = \frac{||\sum_{i=1}^{N_b} M_i a_i - F^{ext}||}{||\sum_{i=1}^{N_b} M_i g||} \tag{3.9}$$

computed at each frame. N_b is the number of body segments, the M_i's are their masses, the a_i's their linear acceleration at a given frame, F^{ext} the sum of the linear external forces measured by the sensors at that frame, and g is the gravity constant. I_{err} represents the consistency of on one hand the masses estimation, the MoCap and the filter process, with on the other hand the sensor measurement. The table 3.2 lists the inertia errors for the McGill sequences and the selection we performed on them for the evaluation. We selected the frames for which the error is less than 10% of BW. The sequence 8 has the highest selection rate and the highest inertia error before and after selection. Therefore, this sequence, which is the only case where uniform weights are better than ours, could be an outlier.



(a) *95% confidence intervals*



(b) *68% confidence intervals*

Figure 3.7: *Comparison of the contact force errors for 3 sets of joint torque weights. The first two sets are obtained from the torque limits of table 3.1, and the third set is a uniform weighting. All values are in percentage of body weight (%BW). The vertical segments represent the confidence intervals at 95% (3.7a) and at 68% (3.7b) arround the mean contact force error.*

Figure 3.8: *Sequence 8 plotted at 4 fps. Each image is shifted toward the right for more lisibility.*

Seq. (#)	Size (# frames)	Inertia error (%BW)	Selection (%)	Inertia error after selection (%BW)
1	333	4.73 ± 3.29	94.9	4.11 ± 2.27
2	253	3.78 ± 2.24	99.6	3.65 ± 2.06
3	309	4.03 ± 2.48	98.4	3.73 ± 2.02
4	289	5.75 ± 3.81	88.9	4.52 ± 2.20
5	357	3.39 ± 2.90	98.3	3.09 ± 1.83
6	321	5.83 ± 4.78	91.0	4.36 ± 2.09
7	221	4.68 ± 3.25	94.6	3.81 ± 1.94
8	233	12.14 ± 8.63	65.2	5.47 ± 2.42
9	249	11.00 ± 9.48	66.3	5.02 ± 2.30
10	425	3.13 ± 2.83	97.4	2.75 ± 1.98
11	349	4.69 ± 3.60	95.7	3.83 ± 2.58
12	553	6.97 ± 6.43	83.2	4.15 ± 2.52
13	369	7.13 ± 7.25	85.9	4.22 ± 1.90
14	529	4.79 ± 3.26	90.7	3.90 ± 2.30
15	553	2.34 ± 1.26	99.8	2.34 ± 1.26

Table 3.2: *Inertia error of each sequences as defined in equation 3.9. The selection includes all frames for which the error is less than 10% of BW. The notation . ± . means the average value plus or minus the standard deviation.*

Seq. (#)	Plantarflexion max torque (N.m)	Difference from 126 (%)
1	111	-12
2	110	-13
3	106	-16
4	141	12
5	139	10
6	138	10
7	125	-1
8	186	48
9	163	29
10	118	-6
11	132	5
12	117	-7
13	144	14
14	125	-1
15	149	18

Table 3.3: *Ankle plantarflexion maximum torques (in N.m) obtained by inverse dynamics without ankle costs. The third column is the difference in percentage from the corresponding literature limit [Robert 2013].*

Although the set of joint torque limits is a major component for optimization-based inverse dynamics, other factors were included in the previous evaluation. In the following we quantify the importance of several parameters other than the joint torque limit set. These parameters are the inclusion or exclusion of the character ankle torques in the torque minimization, the inclusion of torque bound limits as inequality constraints for the quadratic program or not, and the use of either a linear or a quadratic torque limits-to-weight conversion. These two last parameters were included in the above comparison between the method of Robert et al. and our results. Finally an original method to determine the anatomical directions is presented.

The ankle weights The inverse dynamics can be improved in some cases by removing the ankle weights, which is done by setting the corresponding weights to 0 in the $\mathbf{W}_\tau$ matrix of the cost function equation 3.4. To show the consistency of this process, the table 3.3 compares the main ankle axis (plantarflexion[1]) torque maxima obtained without ankle weights, with the standard value of [Robert 2013]. The average signed difference of +6% shows that our results are consistent with standard values. The average max torque value is 134 N.m. The figure 3.9 shows the effect of the ankle weights on the contact force errors. Surprisingly, suppressing

[1] plantarflexion: the foot moves down

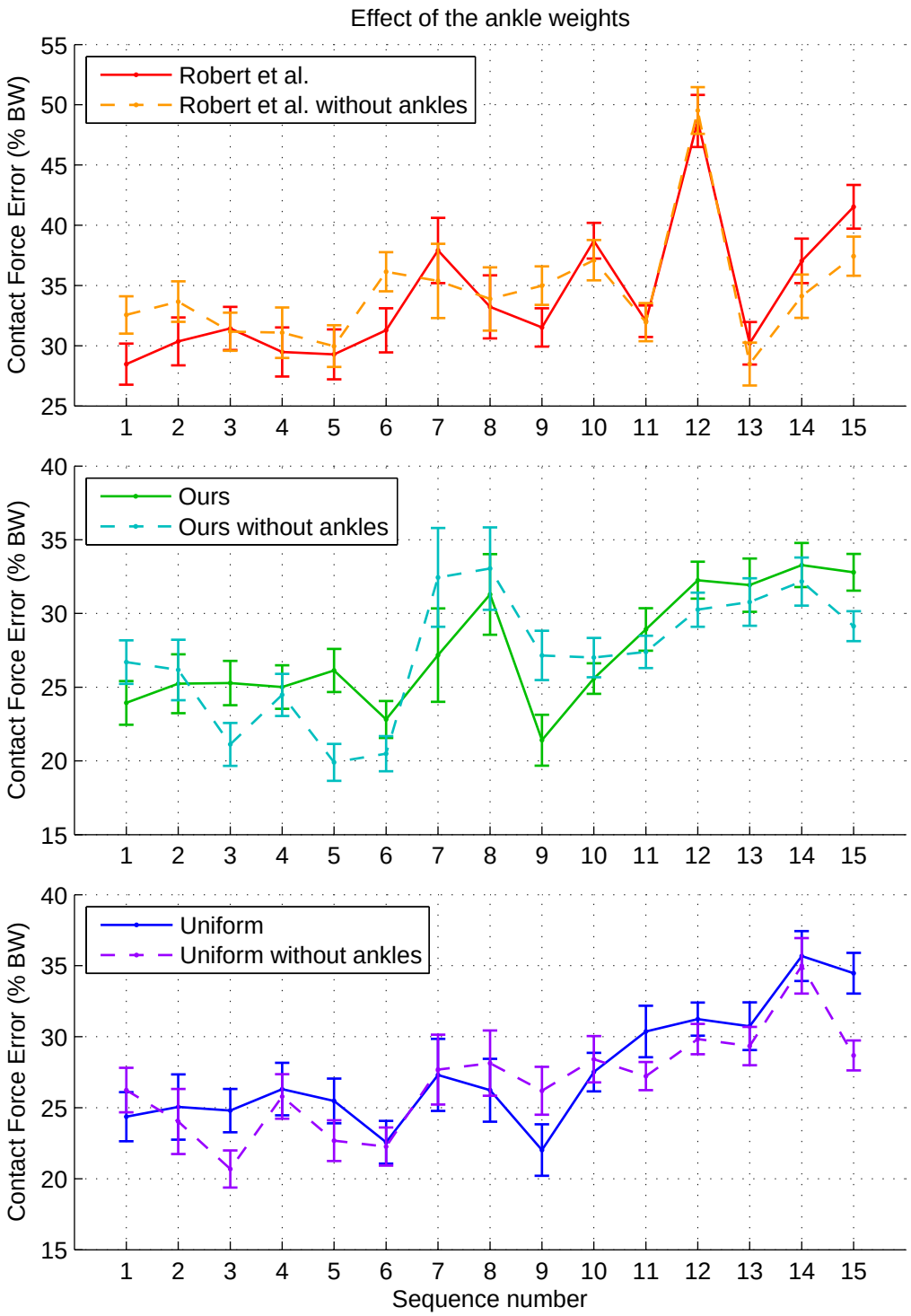
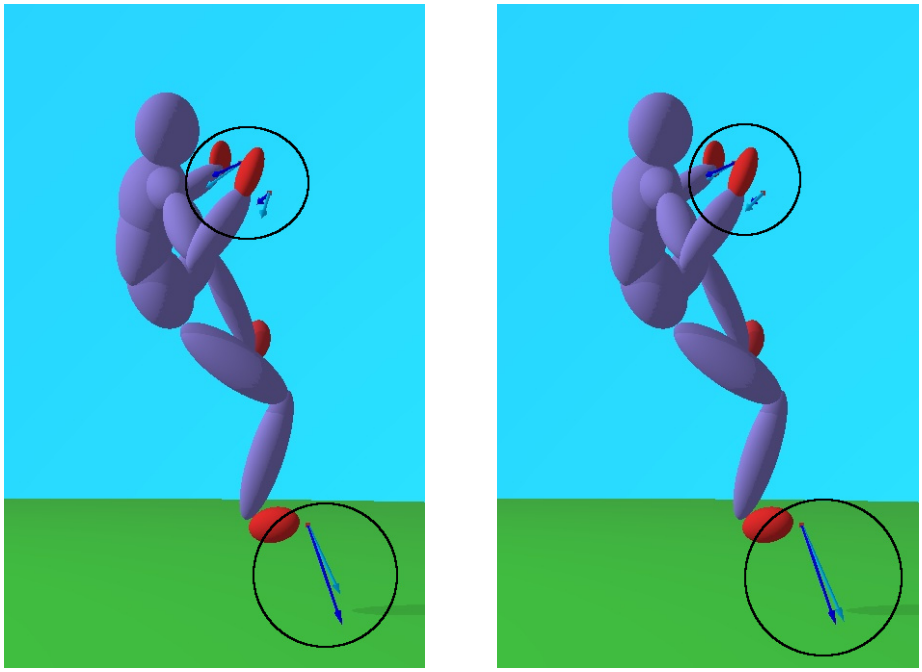


Figure 3.9: *Influence of the ankle torques minimization. The continuous curves are the methods retained for the final comparison. They all include the ankle torque minimization. The dashed curves correspond to results obtained without ankle torque minimization. The confidence intervals, plotted vertically, are at 95%.*

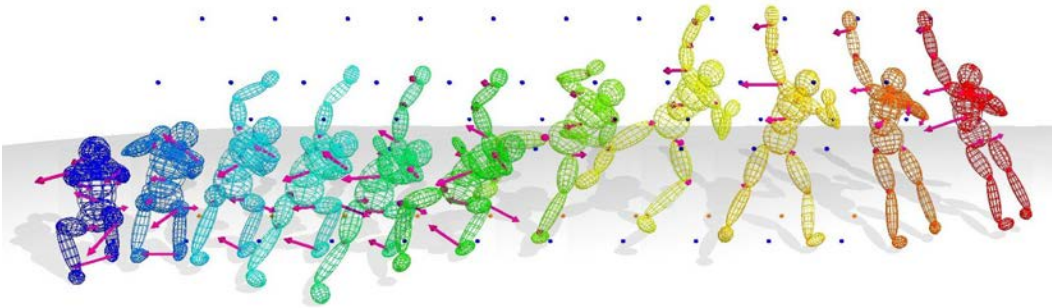


(a) *with ankle weights* **(b)** *without ankle weights*

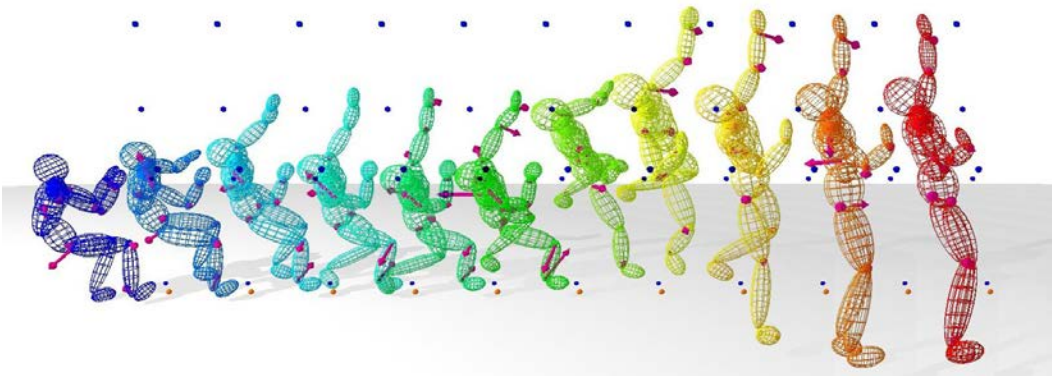
Figure 3.10: *Effect of setting the ankle torque weights to zero on sequence 15 at 3"48. Dark blue arrows are the contact forces measured by the sensors. Light blue arrows are the contact forces estimated by inverse dynamics.*

the ankle cost reduces significantly the error for 3 sequences over 15 for our results and for the uniform weighting. In order to understand these improvements, a frame where it occurs is displayed in the figure 3.10. The motion performed by the climber at this instant was to move his CoM closer to the wall in order to load his right foot and to unload his left hand for reaching the final hold. At this instant, the dorsiflexion[2] maximum angle is reached, leading to the ability for the climber to generate as much passive plantarflexion torques as desired, and therefore bypassing totally the need of minimizing it. Conversely the sequence 9 is always better with ankle torque weights. This sequence is plotted at 4 fps in figure 3.11, with the ground truth torques in magenta. At the beginning (dark blue frame) and at the middle (dark green frame) of this sequence, which are the most torque consuming instants, the torques are almost equally distributed between the right ankle, the right knee, the right hip, the lumbar joint, and the right shoulder, showing that for energetic motions, the limits weighting play a more important role than for easy motions.

[2]dorsiflexion: the foot moves up



(a) *back view*



(b) *right view*

Figure 3.11: *Sequence 9 at 4 fps with ground truth torques, right shifted every frame for visibility.*

Torque bound limits We do not use joint torque bound limits, whereas [Robert 2013] does. The figure 3.12 shows the effect of setting bound limits on joint torques as inequalities in addition to the weighted minimization of the joint torques. These plots shows no significant differences on the resulting contact errors when the computation is done with or without bound limits on the joint torques. The torque limits choosen by Robert et al. are lower than our ground truth joint torque maxima, i.e. computed with sensor data, for the joints the most involved in climbing motions (e.g. shoulder adduction[3], knee extension). The McGill motions thus require torques above standard torque limits, leading to some infeasibilities for the quadratic program with standard bound limits. Therefore we remove bound limits because their only significant effect is to reduce the feasibility of the quadratic program.

Limit-to-weight conversion We evaluated two different methods to convert joint torque limits into joint torque weights. The first one is linear ($weight =$

[3]adduction: the upper limb moves down laterally

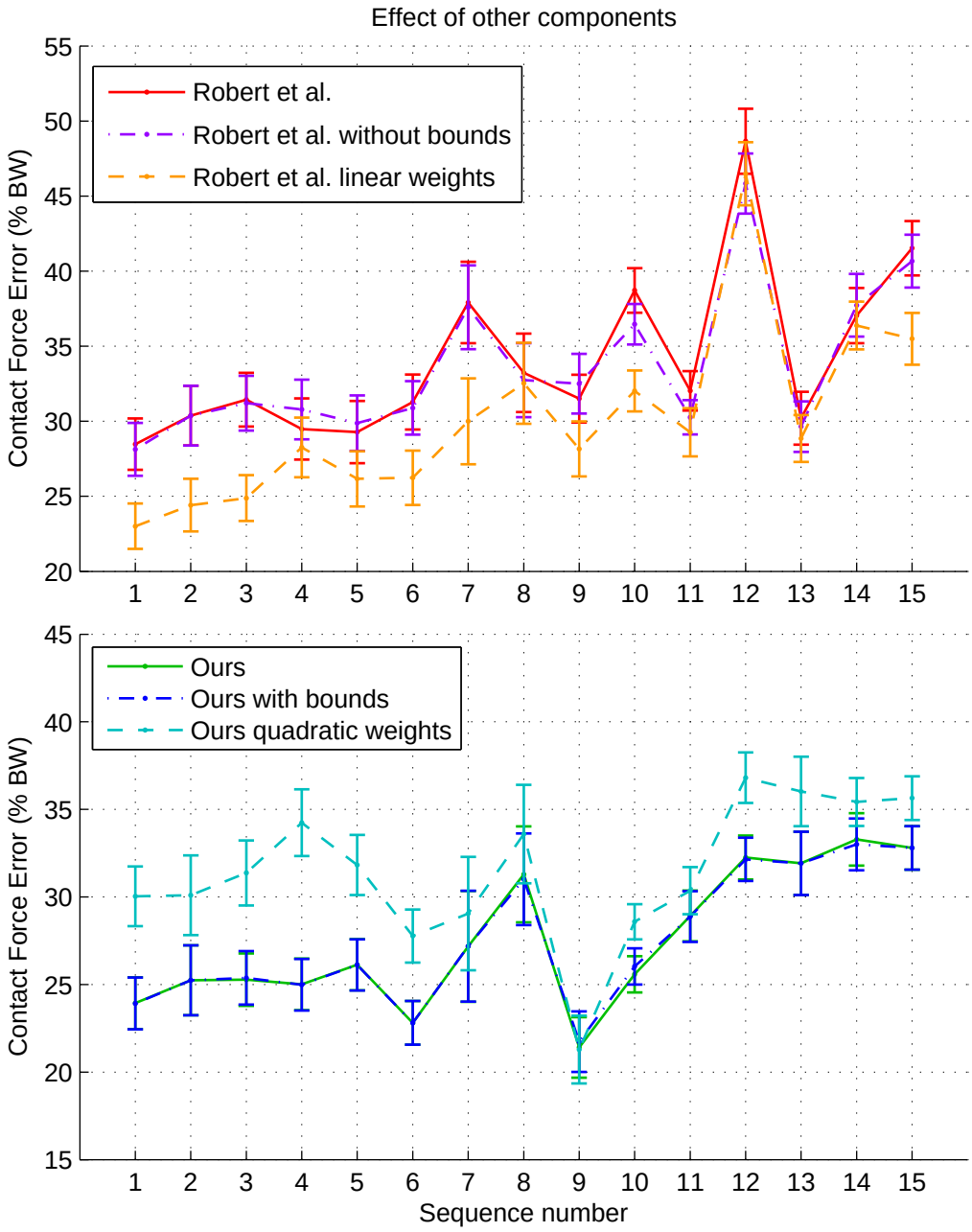


Figure 3.12: *Influence of other factors on the inverse dynamics errors. The two factors tested are the inclusion or not of bound limits on joint torques and the quadratic vs linear limit-to-weight conversion. The vertical segments are the 95% confidence intervals.*

set	Robert et al.	Ours	Uniform
limits	108 ± 68	91 ± 56	100 ± 0.0
linear weights	0.018 ± 0.019	0.035 ± 0.087	0.01 ± 0.0
quadratic weights	0.00068 ± 0.00124	0.00847 ± 0.04283	0.0001 ± 0.0
normalized			
limits	0.1459 ± 0.0915	0.1464 ± 0.0906	0.171 ± 0.0
linear weights	0.121 ± 0.123	0.065 ± 0.161	0.171 ± 0.0
quadratic weights	0.083 ± 0.152	0.034 ± 0.171	0.171 ± 0.0

Table 3.4: *Analysis of limits and weights variations by means and standard deviations (mean ± s.d.). For the first three rows, the limits are in N.m, linear weights are in $N^{-1}.m^{-1}$, and quadratic weights are in $N^{-2}.m^{-2}$. For the last three rows, the quantities are normalized, so there is no unit. The normalization is done at 1 here. As described previously, the final weights are then multiplied by* 10^3.

$1/limit$), and the second one is quadratic ($weight = 1/limit^2$). We use the linear weighting, whereas in [Robert 2013], the quadratic weighting is used. These weightings are evaluated in the figure 3.12, with both standard and McGill limits. The linear conversion improves significantly the results in 17 cases over 30 (with 95% of confidence), and does not change significantly the results for the other cases. This improvement is not due to the fact that the linear conversion makes the weights closer to uniform weights. Indeed, the table 3.4 shows the mean and standard deviation of the three evaluated sets of limits, with the mean and standard deviation of their associated weights with both linear and quadratic limit-to-weight conversions, and the same for the corresponding normalized sets, as the weights are normalized after conversion in our algorithm (see section 3.3.4). The standard deviation of the normalized quadratic weights of Robert et al. is 0.152, whereas the standard deviation of our normalized linear weights is 0.161. This shows that the first weighting is closer to a uniform weighting than the second one, thus the closeness to the uniformity is not the criteria that explains the improvement of the linear conversion. The explanation of this improvement will be illustrated with an abstract example. Let's consider the quadratic program

$$\min_{f_a, f_b} \quad \frac{w_a}{2} f_a^2 + \frac{w_b}{2} f_b^2 \tag{3.10}$$

$$\text{s.t.} \quad f_a + f_b = f_{ext} \tag{3.11}$$

where f_a and f_b are some 1-D actuation forces, w_a and w_b are their associated weights and f_{ext} is an arbitrary force that the actuation forces must reach. Let's consider that the limits of the actuation forces are respectively F_a and F_b. The system 3.10, 3.11 can be solved using Lagrangian multipliers and Mapple, which gives the solution

$$f_a^* = \frac{w_b}{w_a + w_b} f_{ext} \qquad \text{and} \qquad f_b^* = \frac{w_a}{w_a + w_b} f_{ext}. \tag{3.12}$$

The ratio $f_a^*/f_b^* = w_b/w_a$ is thus linearly proportional to the weight ratio. Consequently, normalizing the force by their maxima in equation 3.10, i.e. using a quadratic weighting, gives a final force ratio of $f_a^*/f_b^* = (F_a/F_b)^2$, whereas using a linear weighting leads to a final force ratio of $f_a^*/f_b^* = F_a/F_b$. We conclude that in this case, the linear limit-to-weight conversion better respect the force limit proportions. Respecting the limit proportion is especially useful in the extreme case where $f_{ext} = F_a + F_b$, meaning that all the available forces are needed. With quadratic weights, equation 3.12 becomes

$$f_a^* = F_a^2 \frac{F_a + F_b}{F_a^2 + F_b^2} \qquad \text{and} \qquad f_b^* = F_b^2 \frac{F_a + F_b}{F_a^2 + F_b^2}. \tag{3.13}$$

Except in the degenerated case where $F_a = F_b$, this equation gives that $f_a^* \neq F_a$ and $f_b^* \neq F_b$, but equation 3.11 tells that $f_a^* + f_b^* = F_a + F_b$, so one of the actuation force must exceed its limits. On the contrary, with a linear weighting, equation 3.12 becomes $f_a^* = F_a$ and $f_b^* = F_b$, and the character is able to reach its global limit while respecting its individual actuator limits.

Anatomical directions The anatomical directions are determined using successively two inverse dynamics optimizations. The anatomical weightings presented in table 3.1, are implemented by projection of joint torques on the anatomical directions corresponding to the given weights. This projection is done in the equation 3.4. Using a quadratic term in this equation leads to the lost of the sign of the projected torque component, and has for consequence the confusion between the opposite anatomical directions. Such pairs of directions can have different weights, as it is the case for the knee flexion and the knee extension for instance. To avoid this confusion, two inverse dynamics computations are needed. The first one uses an arbitrary set of weights (the odd lines in table 3.1). The resulting torques are then projected on the anatomical axes, and one direction in each opposite pair of directions is selected. Then the inverse dynamics is recomputed with the new weights. This process is not exact as we observed that between the first and the second optimization, the ankle torque can switch from inversion[4] to eversion for instance, but at least the directions bearing the biggest torques, such as the flexion or extension of almost every joints for instance, are well estimated.

3.5 Conclusion

Following a criteria of minimal torques and contact forces, we showed that the full inverse dynamics problem can be accurately solved from motion data alone, for human activities such as walking or climbing. While not as accurate as force plates, our simple formulation opens the possibility of easily estimating useful dynamic data for motion analysis and physically based animation. It builds upon previous contributions in this area by increasing the accuracy of results that can be expected

[4]inversion: foot internal rotation around its horizontal main axis, the little toe goes down

from such a motion-based approach, and by providing an original evaluation of this method on climbing motions. Our approach works well when the performance itself is actually optimal. In the climbing situation for example, if a subject in a still position on four holds pushes equally on all holds so that his pose does not change, the true non-optimal forces will not be recovered. To summarize, our approach is able to recover dynamics for an optimal usage of forces. This is a limitation but covers most of the cases in standard locomotion. Our work also addresses another issue concerning contact: how contact forces are smoothly established and broken. Currently, our procedural model of transition is based on fixed parameters for amplitude and duration. As force measurement are available from our experimental setup, it will be interesting to automatically learn these parameters and quantify how much they specifically relate to the situation or to the subject's physical characteristics.

Chapter 4

Motion Synthesis by Timing Optimization

Contents

4.1 Introduction **73**
4.2 Related Work **75**
4.3 Problem Formulation **77**
4.3.1 Continuous timing optimization formulation 77
4.3.2 Trajectory sampling to define discrete optimization variables 79
4.3.3 Timing resampling for inverse dynamics 80
4.3.4 Timing cost 82
4.4 Results **87**
4.4.1 Bouldering motion 87
4.4.2 Input parameters sensitivity and setting 96
4.4.3 Long sequence 117
4.5 Conclusion **140**

4.1 Introduction

Human motion synthesis is the art of making a 3D human-like character moving in a virtual environment. Once the character and the environment have been modeled, the next step is to determine the temporal behavior of the character's degrees of freedom with respect this environment, which is the issue addressed in this chapter. Although an artist alone could finely tune each movement of the character with some carefully designed 3D interface, we demonstrate in this chapter how the analysis of physical quantities related to the wanted motion could save hours of works to such an artist. Whereas most of the previous works consider the total duration and sometimes the instants of the different events as input data, we transversally consider that the known quantities are the place where the events occur, and that the unknowns are the instants of these events along with the total duration of the task to perform. The precise case of climbing motions have been chosen, where the particularly restricted environment lets a few choices to the character on the path to follow, and a focus can be made on the timing with which the movements

are performed. A major advantage of optimizing solely on the timing is the low dimensionality of the problem to resolved, enabling a detailed study of the behavior of the proposed algorithm.

The previous approaches in physically-based motion synthesis mainly focus on motions such as walking or running, where the environment is made of a continuous infinite plan. In such environment, two of the most difficult tasks to tackle are on one hand the posture control, which is largely under-constrained due to the multiplicity of the gestures a character can perform on the ground, and on the other hand the balance maintaining, because the polygon of support is made most of the time of one foot during walking or running. Our work tackles a new kind of motions where the environment is made of a sparse set of protrusions vertically distributed called the holds, used for the ascent. This special environment resolved the two previously mentioned difficulties by providing a restricted path to the climber, imposing the gestures to do, and by enabling to have most of the time three contacts with the environment among the hands and the feet, ensuring that the balance can always be maintained.

In addition to kinematic restrictions due to the holds, we moreover assume that the climber tries to reduce his energy consumption to be able to climb as high as possible, without running out of energy. This minimization is especially needed in climbing, where the arms are as mush used as the legs. The upper limbs being the weaker limbs, usually not included in locomotion tasks, minimizing the energy is a necessity in climbing, as opposed to bipedal locomotion, where the lower limbs can be used in various non optimal ways without running out of energy.

Although the climbing activity has already be tackled by roboticians almost a decade ago, the aim of those studies are different than ours. Indeed the driven criteria for the control of multi-legged robots is the stability of the controller, leading to quasi static motions. In our case, we can afford to be close to stability edges by focusing solely on the efficiency of the motion, to produce human-like instead of robot-like motions. The resulting motions are optimal, as observed among trained climbers, whose experimented techniques lead to fluid motions, without hesitation.

Finally, timing optimization is a convenient way to address the human-like character motion synthesis problem, as it brings a simple 1D formulation. This simplicity allows us to tackle more complex motions and longer sequences. It also allows us to analyze finely the output of the optimizer, in order to accurately interpret convergence failures and measure the quality of the results. For example, a comparison is made in this chapter between the segmented optimization technique with the corresponded all-at-once optimization, with the interesting conclusion that the two results differ.

In the following, the related work is reviewed before presenting the formulation of the timing optimization problem. In the section results, we provides two timing optimization examples on bouldering examples where the motion is based on a few MoCap keyposes, on which the convergence of our algorithm is analyzed in details. Then a longer example is tackled, along with the evaluation of the drawbacks and benefits of the segmented optimization.

4.2 Related Work

Physics-based 3D character animation has been studied for more than 25 years. One of the earliest works on this subject [Witkin 1988] defines spacetime constraints for optimizing a trajectory for the Luxo lamp. Given the start pose, the end pose and the contact instants, a trajectory that respects the Newton's second law is found. The main difficulty of this optimization problem is its size, which is the number of control frames times the number of degrees of freedom (DoFs) of the character. Popovic and Witkin [Popović 1999] overcome this issue by predefining the motion on high level body handles. Similarly, high-level motion planning is used by Mordatch et al. [Mordatch 2010] to control locomotion in a varied environment, and by Al Borno et al. [Al Borno 2012] to satisfied varied user objectives. Others directly optimize the full-body controllers for gait cycle on a planar terrain [Wampler 2009], or the optimization parameters for a given motion [Liu 2005]. The tractability of spacetime optimizations can also be obtained with monocular videos to guide the optimization [Wei 2010], or with motion graphs to reduce the search space [Safonova 2008, Ren 2010], or even by using an analytical gradient for the external forces and for the character momentum, as in [Fang 2003]. In this last work, a physically valid motion from a reference motion of a character of 22 DoFs is found, but the whole timing is provided as input, and no internal actuation is computed. In our case, those two last quantities are optimized to increase the realism of the result.

Recently, the spacetime optimization has been generalized by Mordatch et al. [Mordatch 2012] to more complex characters in an open-loop context. In their formulation, the overall movement time T, is partitioned into K phases of $0.5s$. The character state is optimized at phase boundaries, and the overall motion is obtained by cubic spline interpolation of those intermediate states. Thus the resulting motion is spatially optimal, that is to say optimal for a fix time window. Conversely, our method finds the temporal optimum, which is the optimal motion for a fixed spatial trajectory.

The second main approach for physically-based character animation has been historically the design of specific controllers for human, monopedal and quadrupedal locomotion [Raibert 1991, Hodgins 1995, Coros 2011]. If these controllers were gathered in a broad repertoire of lifelike motor skills, Faloutsos et al. [Faloutsos 2001] show how they would be composed to produce longer and more varied animation sequences. A general method to design specific controllers for bipedal locomotion, is formalized with SIMBICON [Yin 2007], which consists of a finite state machine where at each state the character is guided towards a target pose, thanks to proportional-derivative (PD) controllers. Wang et al. [Wang 2010, Wang 2012] use SIMBICON to parameterize a spacetime optimization of the human gait, whereas Vondrak et al. [Vondrak 2012b] combine it with monocular video tracking to control human walking, jumping and gymnastics.

Realtime with physically-based animations can be obtained by sequentially optimizing each frame of the animation individually. By this way, the full character

can track a given motion [da Silva 2008, Macchietto 2009, Liu 2010, Brown 2013], or some body features can be controlled by offline user handles [Abe 2007], that can even be combined with a state machine [Jain 2009, de Lasa 2010]. In comparison with spacetime approaches, the drawback of such methods is the ability to only satisfy instantaneous objectives. Another way to obtain realtime physically-based animation is to combine it with online user interactions, as in [Laszlo 2000]. In this work, simple 2D characters are controlled by PD-controllers, guided by some user commands interfaced with mouse and keyboard. For sensitive motions such as successive jumps with a back-flip for instance, checkpoints are used for the creation of the animation. A more recent example of such realtime interaction is the online game QWOP [Foddy 2008], a game where an Olympian must run a 100 meter race, controlled by the player with the keys Q and W to lift right and left thigh, and O and P to control the calves. Although this kind of interactive controllers provides addictive games, they require hours of training to obtain realistic motions. Thus there use in character animation, where offline tuning of the motion is often used to obtained fine details, is questionable.

As opposed to realtime, the issue of physics-based animation can also be tackled by considering the realism of the model itself on an anatomical point of view. Following this criteria, detailed musculoskeletal models have been designed and animated for simulating the human breathe [Zordan 2006], the neck movements [Lee 2006], the full upper body [Lee 2009], the softness of the skin [Jain 2011], and the lower body [Wang 2012]. In this last work, a realistic human lower limb model is optimized to obtain the activation parameters of a walker and a runner. The optimization is initialized with hand-tuned values based on the work of Geyer and Herr [Geyer 2010]. The gait cycle is made of 4 states, with transitions occurring either at the foot strike instants, or when the horizontal center of mass (CoM) position with respect to the ankles is above a given value. The CoM horizontal target velocity is also given. In our case, the speed of the CoM is optimized instead of given. While very appealing for their anatomical realism, the high complexity of the models presented in this paragraph makes them difficult to control in the context of acyclic and relatively long motion. Moreover, both precise anatomical models and complex activation patterns are available in the case of well biomechanically studied activities such as walking or running, but these models cannot be generalized directly for a drastically different and less studied activity such as climbing.

In robotics, several climbing robots already exist. In [Bretl 2005], a random sampling is used to find feasible climb trajectories for a planar three-limb robot and for a 3D four-limb robot. In [Linder 2005], a planar four-limb climbing robot is controlled by a similar method, with 1D force sensors at the robot end-effectors to control the contact establishment, and with a basic video-based hold detection (the holds contain green LEDs). These three mentioned robots evolve on nearly vertical walls, with quasi static motions. Indeed in robotics, the main required feature is the stability of the robot, and the optimality of its movements is often neglected. Conversely our work focuses on the optimality of climbing motions. A survey on climbing robots [Chu 2010] classifies them into six categories according

to there locomotion mechanism: legged, wheel-driven, tracked, translation, cable-driven and combined; and into five categories according to their adhesion device: suction, magnetic, gripping, railguided, and biomimetic. If our simulated climber were a robot, it would fall in the categories legged and biomimetic.

Finally, it is interesting to notice the similarity of our work with the work of McCann et al. [McCann 2006], where an input MoCap motion is retimed by changing the total duration of the sequence. The key difference here, is that in their case, the total duration of the animation is an input parameter, whereas in our case, the duration is optimized. Note that this duration is given in most of the synthesis works mentioned above. However, this parameter is crucial in motion synthesis, because it acts upon the whole animation by constraining the average velocity to perform the given tasks. Moreover, the optimization of this parameter is not as obvious as it could appear at first sight. Indeed, the smoothest motion obtained with time bounded optimization by minimizing for instance average squared joint torques or accelerations, is not anymore a good objective as it leads to infinitely long animations. Another major difference with [McCann 2006] is that they are focusing on a fast computation of almost single-contact motions, whereas we are focusing on the accuracy of highly multi-contact motions. Therefore, their resolution of the multi-contact inverse dynamics problem consists solely on feasible contact forces, and thus an interpolation of some scatter solutions is enough to obtain a continuous range of solutions used for the retiming. In our case, we do not rely on a fixed set of precomputed contact forces, but directly on the minimization of joint torques with fully varying contact forces, which is particularly needed to obtain energy efficiency in highly multi-contact contexts. Finally, the main input of [McCann 2006] is a full MoCap sequence, whereas only sparse end-effector positions are needed in our case.

4.3 Problem Formulation

4.3.1 Continuous timing optimization formulation

This section describes the context of the timing optimization. The inputs are the geometry and inertial properties of a climber, and the bouldering problem or the route to climb, described either by a series of keyposes of the climber, of by a sequence of holds to take. The outputs are a realistic continuous climbing path to follow, as well as the optimal timing along this path that minimizes the joint torques needed for the ascent. The timing optimization is summarized figure 4.1.

We have a character with $n_j \in \mathbb{N}$ joints. In practice, we have $n_j = 18$ joints, including 4 hinge joints, 1 root joint and $n_j - 5$ ball joints. Each non-root joint connects two rigid bodies together, forming an articulated tree of rigid bodies. The character evolves in the generalized coordinates space $\mathbb{G} = \mathbb{R}^d$. In practice, $d = 49$ $(= 3 \times (n_j - 5) + 4 + 6)$.

The character's trajectory is an interpolation of keyposes in $\mathbb{G}$. This interpolation is a composition of linear interpolation, inverse kinematics, 1D cubic splines

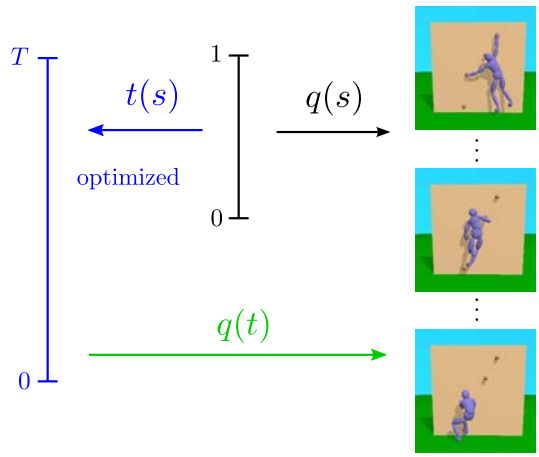


Figure 4.1: *Problem formulation. The aim is to optimize for the timing, $t(s)$, defined on the parametric space $[0,1]$, knowing the parametric curve, $q(s)$, representing the spatial climbing path, with no timing. The parametric coordinate, s, evolves in $[0,1]$. $q(t)$ represents the timed trajectory of the climber.*

and quaternion cubic splines. The interpolation is described by the function

$$\begin{array}{rccc} q: & [0,1] & \longrightarrow & \mathbb{G} \\ & s & \longmapsto & q(s), \end{array} \tag{4.1}$$

where s is the parametric coordinate of the trajectory. $s = 0$ means that the climber is at the beginning of the climbing path, and $s = 1$ means that the climber is at the end of the path. $q(s)$ represents the character joint configuration associated with a given parametric coordinate s.

The originality of our work is to consider that the time is evolving in the parametric space, and that the mapping between the parametric coordinates and the generalized coordinates is fixed and independent of the time. Thus the time is a function of the parametric coordinates, which we can write

$$\begin{array}{rccc} t: & [0,1] & \longrightarrow & [0,T] \\ & s & \longmapsto & t(s). \end{array} \tag{4.2}$$

The overall problem is to find this timing function $t(s)$. This involves finding the total time of the climbing run T, and the mapping between parametric coordinates and temporal points of $[0,T]$. This objective is depicted in the figure 4.1. For the function $t(s)$ to be well defined, we assume that the final parametric coordinates are increasing with the time.

The main difficulty for implementing the timing optimization described above, is to define the samplings for the parametric space $[0,1]$ and for the timing space $[0,T]$, from which the optimization variables are defined. There are four levels of sampling, outlined here and then detailed later.

- **Level 1:** Input keyposes parametric sampling, which is set once for all before the optimization.

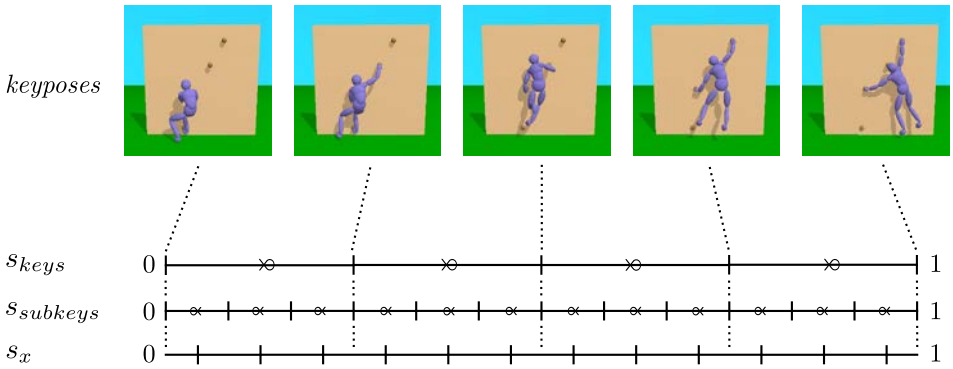


Figure 4.2: *Definition of the parametric coordinates for the input keyposes (s_{keys}), for the virtual sub-keyposes ($s_{subkeys}$), and for the optimization variables (s_x). Here, $N_{keys} = 5$ and $N_x = 12$.*

- **Level 2:** Parametric sampling for the optimization variables, which is fixed in the parametric space, but the associated points of the timing space are optimized.
- **Level 3:** Temporal sampling for the inverse dynamics computation, which is regularly spaced in the temporal space, with a number of samples fixed once for all before the optimization.
- **Level 4:** Temporal sampling for the final rendered animation, which is regularly spaced in the temporal space, with a varying number of samples, in order to match the rendering framerate.

The keypose sampling and the optimization variable sampling are detailed in the section 4.3.2. The inverse dynamics sampling is described in the section 4.3.3. The final rendering sampling is not detailed. Finally the section 4.3.4 presents the cost function evaluation, computed on the inverse dynamics samples.

4.3.2 Trajectory sampling to define discrete optimization variables

The keyposes, defining the climb trajectory by interpolation, are also called keyframes, or keys. Their number is noted N_{keys}. The parametric coordinates of the keyposes, noted s_{keys}, is set before the optimization as a regular sampling between 0 and 1. The figure 4.2 illustrates this sampling when $N_{keys} = 5$. To refine the timing optimization, we add several virtual sub-keyposes between the original input keyposes. Their number is noted $N_{subkeys}$. The parametric coordinates of the sub-keyposes, noted $s_{subkeys}$, are also regularly sampled.

The aim of the timing optimization is to find the timing function $t(s)$. This function is discretized at each sub-keyposes, such that the problem becomes to know the time instants at which the climber goes through each sub-keypose. For

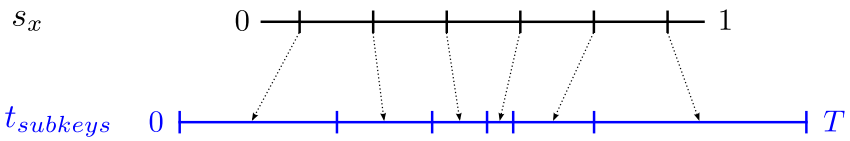


Figure 4.3: *The optimized quantities, or optimization variables, noted x, are the time intervals between sub-keyposes (in blue). In this example, the number of unknowns for the optimization, N_x, is 6. This is also the size of s_x, which are the parametric coordinates associated to x.*

the timing function $t(s)$ to be well defined, we need that the parametric coordinate s increases when the time t increases, such that the function from t to s is bijective, and thus $t(s)$ exists and is an increasing function of s. To ensure this last property along the sub-sampling, we take as optimization variables x, the time intervals between two successive parametric coordinates in $s_{subkeys}$, and constrain x to be positive. The parametric coordinates related to x, noted s_x, are therefore in-between the parametric coordinates of the sub-keyposes. The size of x, noted N_x, is thus the number of sub-keypose intervals $(N_{subkeys} - 1)$.

To summarize, the input keyposes are refined with intermediate sub-keyposes. The optimization variables are defined as the time duration between two successive sub-keyposes, as represented figure 4.3. The convertion from the optimization variable $x = (x_1, \ldots, x_{N_x})$ to the timing $t_{subkeys} = (t_0, \ldots, t_{N_x})$ is obtained by

$$t_{subkeys}: \qquad t_{i+1} = \begin{cases} 0 & \text{if } i = 0, \\ t_i + x_i & \text{otherwise,} \end{cases} \qquad \forall i \in 0..N_x. \tag{4.3}$$

Thus the total time of the sequence

$$T = \sum_{i=1}^{N_x} x_i, \tag{4.4}$$

which is also t_{N_x}, is thus implicitly included in the optimization.

4.3.3 Timing resampling for inverse dynamics

The inverse dynamics operates on constant time intervals, and our cost function is based on an inverse dynamics computation to evaluate the cost of the climb. Therefore, a resampling of the subkeys timing is needed to convert the time-varying pairs $\{s_{subkeys}, t_{subkeys}\}$, into time-regular pairs $\{s_{id}, t_{id}\}$, that is to say pairs sampled at a regular time step. An example is shown figure 4.4.

The number of inverse dynamics frames, N_{id}, is constant to ensure the continuity of the cost function, and is set before the optimization. This number is used with the total time T of the trajectory obtained by summing the optimization variables (equation 4.4), to obtain the timestep for the inverse dynamics

$$\Delta_{id} = \frac{T}{N_{id} - 1}. \tag{4.5}$$

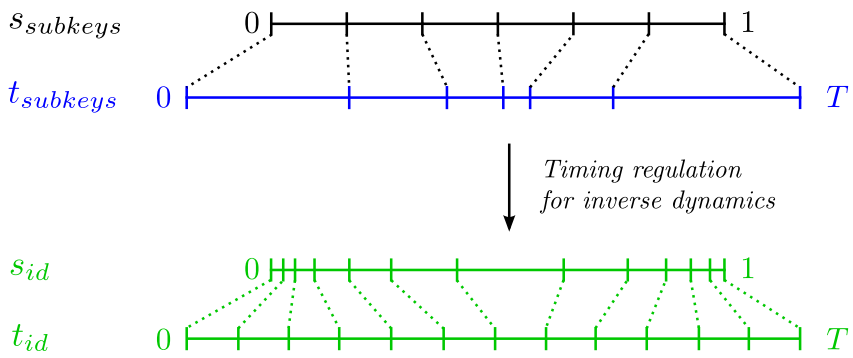


Figure 4.4: *The resampling of the parametric-regular pairs* $\{s_{subkeys}, t_{subkeys}\}$*, into the time-regular pairs* $\{s_{id}, t_{id}\}$*. Here, the optimization size* $N_x = 6$*, which equals the size of* $s_{subkeys}$ *minus 1, and the number of frames for the inverse dynamics* $N_{id} = 13$*, which equals the size of* t_{id}*. The green color represents quantities for the green path of the figure 4.1, where the parametric coordinates are used as intermediate quantities to go from the time space to the character pose space.*

Note that Δ_{id} is not the final time step of 25 fps or 100 fps, but is a time step varying during the optimization. Ideally, the cost function should be a continuous function of N_{id}, such that the final time step could be used directly, but this feature is let for future works.

Once the inverse dynamics timestep is defined, the inverse dynamics timing t_{id} is computed with the relation

$$t_{id}: \qquad t_k = (k-1)\Delta_{id}, \qquad \forall k = 1..N_{id}. \tag{4.6}$$

t_{id} is then used to obtain the related parametric coordinates s_{id}, by a cubic spline interpolation of the pairs $\{t_{subkeys}, s_{subkeys}\}$. The corresponding Matlab command is

$$s_{id} = \text{spline}(t_{subkeys}, s_{subkeys}, t_{id}). \tag{4.7}$$

For details about this 1-D cubic spline interpolation, one can look at the online documentation of the Matlab's spline function. An illustration of this function is provided figure 4.4.

Finally the trajectory for the inverse dynamics is computed by interpolating the keyframes in the generalized coordinate space

$$q_{id} = \text{ikqspline}(s_{keys}, q_{keys}, s_{id}). \tag{4.8}$$

The subkeys are not needed anymore, as they are themselves defined from the interpolation of the keyframes. The ikqspline function consists of two steps. First a cubic spline interpolation is used for the hinge joints and for the root translations, and a quaternion cubic spline interpolation is used for the ball joints. Then the world coordinates of the contact points on the hands and feet are linearly interpolated.

Finally, the limbs are adjusted procedurally to match the contact points by inverse kinematics. The quaternion cubic spline interpolation is implemented from the work on quaternion cubic spline by J. McEnnan [McEnnan 2003]. By using a procedural inverse kinematics, guided by cubic spline interpolation, we ensure that the hands and the feet do not get into the wall (at least when it is concave), and do not move if they are identical on two successive keyframes. Moreover, the obtained trajectory is continuous, which would not be the case with a classic nonlinear inverse kinematics, producing independent local minima along the trajectory. We have thus defined q_{id}, which are the frames on which the timing cost function is computed.

4.3.4 Timing cost

The timing cost consists of both the actuation needed to follow the climbing trajectory q_{id} with the timing t_{id}, and a contact term ensuring that the current timed trajectory can be done without using contact forces when contact points are not touching the wall. The actuation term, based on joint torques, represents the effort needed to perform the motion. The joint torques computation is presented in the subsection "torques", which provides some complements to the chapter 3. The subsection "contact cost" tackles the issue of the plausibility evaluation of the contact forces. A final subsection "nonlinear objective" is dealing with the proper definition of the nonlinear cost function.

Torques

The first step for the inverse dynamics computation is the velocity and acceleration computations. This computation is done by finite difference of the trajectory frames q_{id}. The associated velocities, $\dot{q}_{id}$, are computed by backward difference, and the accelerations, $\ddot{q}_{id}$, are computed by central difference. We set the undefined bounds, which are the first velocity and the last acceleration, to zero. In practice, the finite difference is perform in maximal coordinates to simplify the derivation of the Coriolis and centrifugal terms.

In a next step, the known quantities of the equation of motion are gathered in a single vector

$$h = M\ddot{q}_i + C(q_i, \dot{q}_i), \tag{4.9}$$

where M is the generalized mass matrix and C gathers the Coriolis, centrifugal and gravitational terms. This known h vector is then projected to isolate its root part h_r, which is not actuated, and its actuated part h_τ. These two vectors enables us to write on the one hand, the joint torques τ as a linear function of the contact forces f_c, and on the other hand, the only equality constraint on the contact forces, which is of dimension 6.

$$\tau(f_c) = h_\tau - J_\tau^T f_c \tag{4.10}$$

$$h_r = J_r^T \ f_c \tag{4.11}$$

where J_r^T transfers wrenches from contact points to the root frame (which can be actually any frame, including the world origin for instance), and sums these transfered wrenches. J_τ^T transfers wrenches from contact points to joint locations, and project them on the actuation axes. They consists of any triplet of orthogonal axes for ball joints, and the hinge axes for hinge joints. A straightforward future work would be to use anatomical axes, as in chapter 3. Note that the anatomical axes are not orthogonal in general, as they are defined by successive Euler angles. Thus an actuation redundancy occurs when using anatomical axes, which does not occur with the orthogonal axes we use in this chapter. Nevertheless, the anatomical axes provide a way to interpret the computed torque components, which can be used by an external user to weight them for the actuation cost evaluation. Ideally, the redundancy of anatomical axes should be avoided by nonlinear inverse dynamics, instead of quadratic based inverse dynamics. The second one, less accurate but faster, could eventually be used as initialization of the first one.

In a final step, by building on two previous works on inverse dynamics [Wei 2010, Mordatch 2013], we obtain the contact forces f_k needed to produce the trajectory by solving at each frame k from 1 to N_{id}, the quadratic program

$$f_k = \underset{f_c}{\operatorname{argmin}} \quad \tau(f_c)^T W_\tau \tau(f_c) + f_c^T W_k f_c \tag{4.12}$$

$$\text{such that} \quad h_r = J_r^T \; f_c \tag{4.13}$$

$$F_c f_c \leq 0, \tag{4.14}$$

where we inject in (4.12) the force-torque linear relation (4.10) to end up with only the contact forces as unknowns. In our case, with a single contact point per hand and per foot, we end up with only 12 unknowns. Taking also τ as unknown would have increased this number of unknowns by 49. The joints weights W_τ are set such that weaker joints, such as the wrists or the neck, are more costly than the stronger ones, such as the leg joints. W_τ does not depend on the frame number, whereas the contact weights W_k are computed at each frame k, as explained in the next subsection. F_c is a friction cone. To reduce the number of variables, we do not use the classical friction basis vectors for F_c, which would imply to have one contact force component per cone generator vector. Instead, F_c projects the contact forces f_c on the outward normals of the facets of the polygonal friction cones. This enables us to use up to 8 facets per cone, without affecting the size of the cost function to minimize (4.12). From the optimal contact forces f_k, the corresponding per frame joint torques τ_k are retrieved with the equation 4.10.

Contact cost

The contact cost W_k is used twice in our optimization framework. On one hand, it is used frame by frame to computed the best contact forces, independently at each frame k, given the current timing. This usage has been presented in the last subsection. On the other hand, the contact cost of the whole trajectory is used to decide how the timing should evolve to give a more plausible motion. This will be

presented in the next subsection. This subsection focuses on the component of the contact cost that guides both level of optimization (per frame, and on the whole trajectory) to determine the final contact forces, which is the per-frame contact weight matrix W_k. This matrix tells the optimizers if a contact point is allowed to generate forces, or not.

The major difference between the contact weighting of the chapter 3, focusing solely on inverse dynamics, and the one used in this chapter, also dedicated to motion synthesis, is that the first one relies on contact time instants, whereas the second one relies on spatial contact kinematics, and is thus independent of time instants. Indeed, the inverse dynamics of chapter 3 applies a smooth contact transition just after the contact establishment and just before the contact breaking. During this transition, the contact point does not move (up to MoCap noise). Conversely here, the contact transitions are done on the spacial localization of the contact point, just before the contact establishment and just after the contact breaking. Despite less accurate, this process is straightforward to implement, and is thus used in state-of-the-art physics-based motion synthesis [Mordatch 2013], but should be changed in future works to increase the contact model accuracy.

We set $n_c = 4$ contact points locally fixed on the character, one per hand and one per foot. We call them local points. n_h holds can be used for the climb. These holds are the environment contact points, and are represented by one 3D point each. There are N_{id} frames. For each frame k, for each local point i, of world coordinates $p_{k,i}$, the distance between $p_{k,i}$ and the environment, noted $d_{k,i}$, is computed as the closest distance between the local point and the set of holds, which gives

$$\forall k \in 1..N_{id}, \forall i \in 1..n_c, \qquad d_{k,i} = \min_{j \in 1..n_h} ||p_{k,i} - p'_j||, \tag{4.15}$$

with $p_{k,i}$ the world coordinates of the local point on the climber, and p'_j the hold coordinates. The local point velocity, $v_{k,i} = \dot{p}_{k,i}$, is computed by finite difference. The contact availability, noted $\alpha_{k,i} = \alpha(d_{k,i}, v_{k,i})$, is expressed by a scalar value in $[0, 1]$. 1 means an available contact, i.e. a local point on the climber that can generate forces, and 0 means a local point that can not be used as contact. The availability is a function of the distance and the velocity of the local point. Those quantities are converted from $[0, +\infty[$ to $[0, 1[$ with the sigmoid function

$$\sigma(x) = \frac{1 - \tanh(k_1 x - k_2)}{2}, \tag{4.16}$$

where x can be either distance or velocity, and $k_1, k_2 \in \mathbb{R}$ a some scalar constants detailed just after the next equation. The contact availability is then computed as a product of both the distance sigmoid and the velocity sigmoid

$$\alpha_{k,i} = \alpha(d_{k,i}, v_{k,i}) = \sigma(d_{k,i})\ \sigma(v_{k,i}). \tag{4.17}$$

The constants k_1 and k_2 are set to 30 and 3 respectively, such that α has the shape plotted in the figure 4.5. With this shape, if the local point is at 1 cm from a hold, with a velocity of 1 cm.s^{-1}, then $\alpha = 0.99$, which means that the contact is

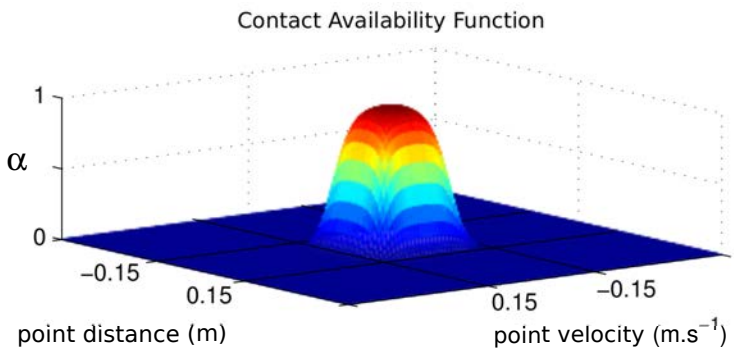


Figure 4.5: *The availability function α turns off contact forces if the potential contact point, a local point on the climber, is far from the environment, or if this point is moving.*

established. If the distance is 10 cm and the velocity 10 cm.s^{-1}, then $\alpha = 0.25$ and the contact is approximately not established. Finally, at 15 cm of distance and 15 cm.s^{-1} of velocity, $\alpha = 0.002$, i.e. the contact is definitely not established.

The contact availability value is then converted into weight for the objective functions. The range of the contact weights is arbitrarily set to $[10^{-4}, 10^{4}]$. A high contact weight value means a costly contact, i.e. a contact that should not be used because it does not exist in practice. Conversely low values are set for contacts that can freely give forces to the climber. The availability-to-cost conversion is implemented with the logarithmic barycentric interpolation

$$w_{k,i} = 10^{-4\alpha_{k,i}+4(1-\alpha_{k,i})}, \tag{4.18}$$

valid for each frame k, and for each local point i. To be able to use these weights with the contact forces in the quadratic form $f_c^T W_k f_c$, the are gathered per frame and rearranged in a diagonal matrix :

$$W_k = \left[\begin{array}{ccc|ccc|ccc|ccc}
w_{k,1} & 0 & 0 & & & & & & & & & \\
0 & w_{k,1} & 0 & & 0 & & & & & & & \\
0 & 0 & w_{k,1} & & & & & & & & & \\
\hline
 & & & w_{k,2} & 0 & 0 & & & & & & \\
 & 0 & & 0 & w_{k,2} & 0 & & 0 & & & & \\
 & & & 0 & 0 & w_{k,2} & & & & & & \\
\hline
 & & & & & & w_{k,3} & 0 & 0 & & & \\
 & & & & & & 0 & w_{k,3} & 0 & & 0 & \\
 & 0 & & & & & 0 & 0 & w_{k,3} & & & \\
\hline
 & & & & & & & & & w_{k,4} & 0 & 0 \\
 & & & & & & & 0 & & 0 & w_{k,4} & 0 \\
 & & & & & & & & & 0 & 0 & w_{k,4}
\end{array}\right] \tag{4.19}$$

Each triplet of columns corresponds to the x-, y- and z-axis components of the contact force generated at a given local point on the climber, at a given frame k.

For instance the first triplet corresponds to the right foot contact force and the last triplet corresponds to the left hand contact force.

Nonlinear objective

The nonlinear energy minimization is the second optimization level. As opposed to the first one, which operates frame by frame, this one operates on the whole trajectory to optimize for its timing. Like the previous one, the objective function to minimize contains both joint torques, modelizing the internal energy consumption of the climber, and contact forces, modelizing the plausibility of the motion.

At a frame k, the joint actuation is a vector noted τ_k. This vector has $d = 49$ components, one component per generalized coordinate. These components are noted with the index $j \in 1..d$. In other words

$$\tau_k = \left[\begin{array}{cccccc} \tau_{k,1} & \tau_{k,2} & \dots & \tau_{k,j} & \dots & \tau_{k,d} \end{array} \right]^T. \tag{4.20}$$

The joint torque consumption is evaluated per component by squared integration of its value, $\tau_j(t)$, over the climb time t. The joint torque consumption on the whole trajectory for component j, noted $\tilde{\tau}_j$, is then

$$\tilde{\tau}_j = \int_0^T \tau_j^2(t)dt \tag{4.21}$$

$$\approx \Delta_{id} \sum_{k=1}^{N_{id}} \tau_{k,j}^2. \tag{4.22}$$

$\tau_j(t)$ is the continuous joint torque function of time for component j. Δ_{id} is the inverse dynamics timestep as defined in equation 4.5. The joint torque consumptions per component are then rearranged in one vector and weighted, to give the actuation consumption

$$\tilde{\tau} = W_\tau \left[\begin{array}{cccccc} \tilde{\tau}_1 & \tilde{\tau}_2 & \dots & \tilde{\tau}_j & \dots & \tilde{\tau}_d \end{array} \right]^T. \tag{4.23}$$

This actuation consumption is the first part of the nonlinear objective function.

The second part of the nonlinear objective function is the air-contact consumption, which indicates the amount of contact forces that are generated at the climber's points that do not touch the wall. This is done by multiplying the contact forces by 0 if the contact is established, or by a positive value if it is not. We found empirically that a range of $[0, 1]$ gives better results than a range of $[10^{-4}, 10^4]$. Consequently we use $1 - \alpha$ instead of W_k to penalize the forbidden forces. At the frame k, there are $N_c = 3 \times n_c = 12$ contact force components, that we index with the letter l, such that

$$f_k = \left[\begin{array}{cccccc} f_{k,1} & f_{k,2} & \dots & f_{k,l} & \dots & f_{k,N_c} \end{array} \right]^T. \tag{4.24}$$

f_k is defined in equation 4.12. The conversion from contact component index $l \in 1..N_c$ to contact index $i \in 1..n_c$ is ensured by the function $i(l) = floor(\frac{l-1}{3}) + 1$.

Similarly to the actuation consumption, the air-contact consumption, noted $\tilde{f}$, of components $\tilde{f}_l$, is integrated over the climb time, which gives

$$\tilde{f}_l = \int_0^T (1-\alpha_{i(l)}) f_l^2 dt \tag{4.25}$$

$$\approx \Delta_{id} \sum_{k=1}^{N_{id}} (1-\alpha_{k,i(l)}) f_{k,l}^2, \qquad \forall l \in 1..N_c \tag{4.26}$$

and

$$\tilde{f} = \left[\begin{array}{ccccc} \tilde{f}_1 & \tilde{f}_2 & \dots & \tilde{f}_l & \dots & \tilde{f}_{N_c} \end{array} \right]^T. \tag{4.27}$$

$\alpha_{i(l)}$ and f_l are the continuous function associated respectively to the discretized functions $\alpha_{k,i(l)}$ (equation 4.17) and $f_{k,l}$ (equations 4.24).

Finally the optimal timing is found by minimizing the norm of the vector containing both $\tilde{\tau}$ and $\tilde{f}$, which gives

$$\min_x \left\| \left[\begin{array}{c} \tilde{\tau} \\ \tilde{f} \end{array} \right] \right\|. \tag{4.28}$$

The solution to this program is a series of interval times, as explained in the figure 4.3. A resampling similar to the one presented in the figure 4.4 must then be performed to obtain a regular timing of the final video timestep.

4.4 Results

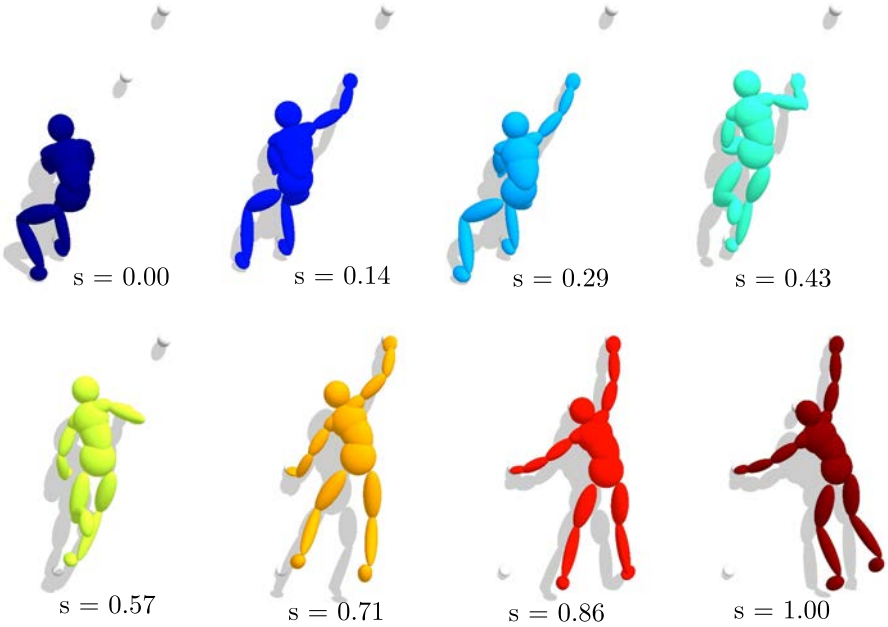
The timing optimization has been applied in two cases presented in the following. The first case presents two short bouldering sequences where keyposes are computed from MoCap data, and the second case presents a longer sequence where keyposes are computed by inverse kinematics. For each case, a detailed analysis of the input parameters is provide. The input parameter analysis constitutes the main part of this section. Finally, the longer sequence evaluation ends up with a video-based ground-truth comparison.

4.4.1 Bouldering motion

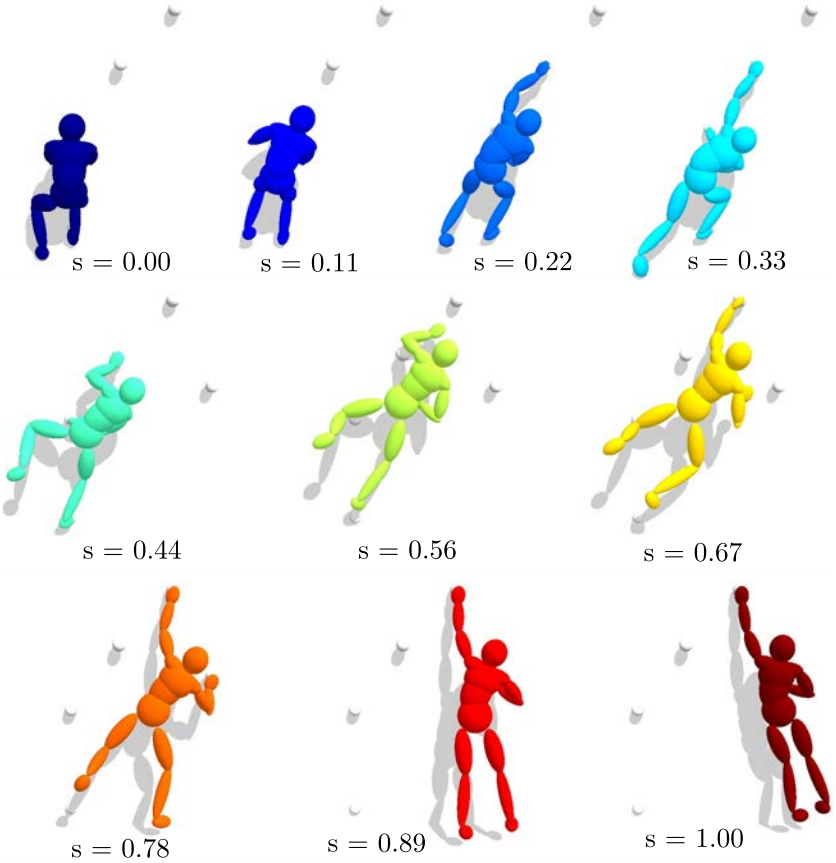
This first series of results shows the application of the optimal timing for bouldering motions containing ballistic phases with energetic impulse phases. We chose to reproduce two sequences of the McGill database such that ground truth timings and keyframes are available. Note that we do not use whole MoCap sequences as input. Instead, only a few keyframes a taken, that could be easily replaced by keyframes modeled by an artist using any standard 3D modeling software. The MoCap timing is not used as input, but only for the evaluation.

Input keyframes and contacts

The figure 4.6 presents the keyframes used for the bouldering examples. The input keyframes controlling the path of the climb have been chosen such that the interpolated path is as close as possible to the path of the original MoCap sequence. Those



(a) *Keyframes for the example 1, from McGill 8. (see figure 2.17)*



(b) *Keyframes for the example 2, from McGill 9.*

Figure 4.6: *The input keyframes for the two bouldering examples and their parametric coordinates. To obtain those poses, first a set of poses is extracted from the MoCap data. Second, the limbs of the climber are adjusted by inverse kinematics for the local contact points to match the holds.*

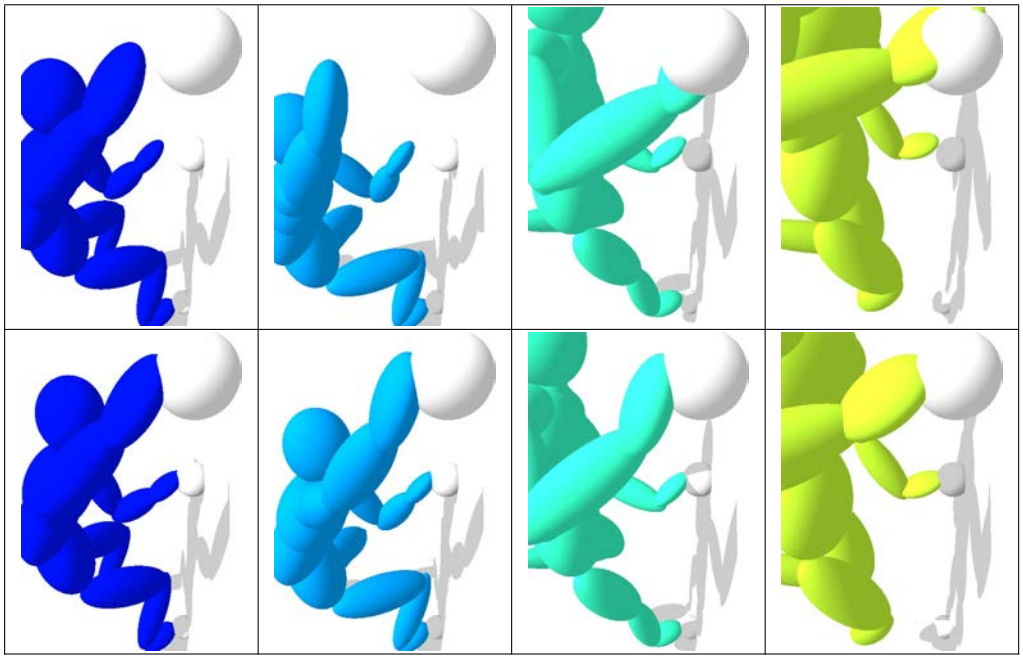
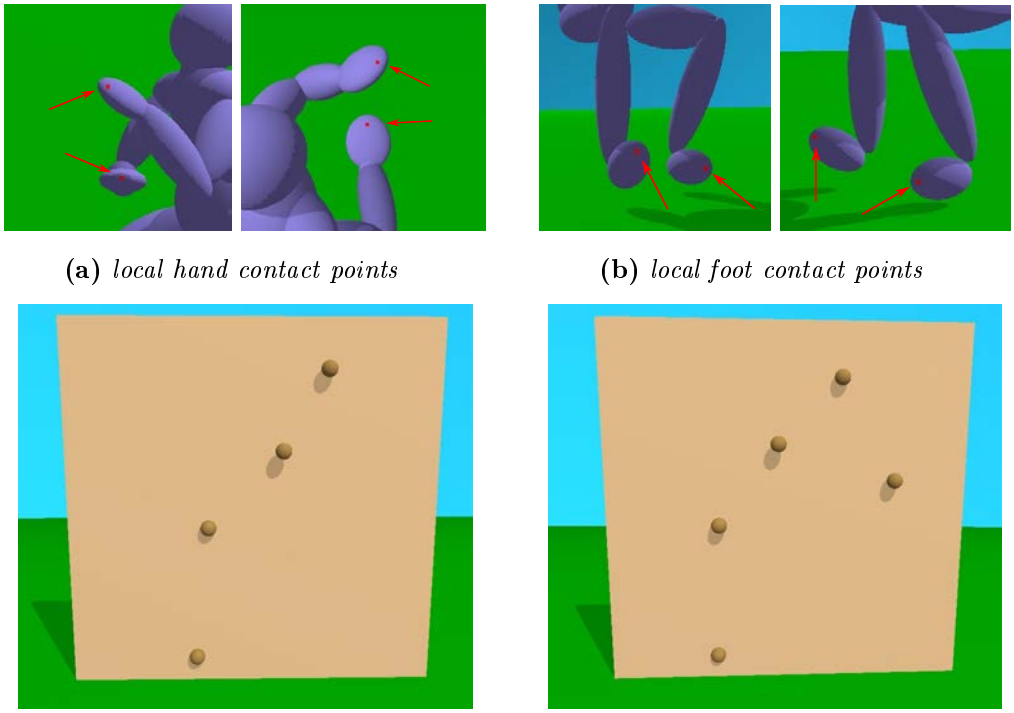


Figure 4.7: *Pre-process of the input data by inverse kinematics. Top row, the MoCap output. Bottom row, the hands and the feet have been adjusted on the holds. We use a full-body iterative inverse kinematics to stay as close as possible to the input data.*

keyframes can be interpreted as the extrema of the climber's path. The figure 4.7 shows the preprocess of the input keyframes. The keyframe generation is done in two steps. The first step is the extraction of the MoCap poses wanted to build the keyframes. The second step is the adjustment of the climber's limbs to match the holds when a contact is supposed to be established. Indeed, due to MoCap imprecision, the hands and the feet of the climber are skating on the holds instead of staying still during a contact. The local contact points on the climber's end effectors are therefore replaced exactly on the holds if they are supposed to be in contact.

The potential contact points are presented figure 4.8. The first set of contacts, the red dots of the figures 4.8a and 4.8b, are the local contact points on the climber. They are fixed on the character end effectors, namely the hands and the feet, and move with them. The second set of potential contact points are the holds available for the climb, represented by the brown spheres of the figures 4.8c and 4.8d. Each hold is modelized by only one potential contact point. This contact point has been estimated by averaging the positions of the local points in the original MoCap sequences. They are localized on the surface of the brown spheres, and are the farthest point from the wall on this surface. Both local and global contacts are continuously present during the whole timing optimization and on the whole sequence. The fundamental difference between local and global contacts is that only local contacts can generate forces on the character, at any time, but forces without



(a) *local hand contact points* **(b)** *local foot contact points*

(c) *world contacts for example 1* **(d)** *world contacts for example 2*

Figure 4.8: *The input contacts for the bouldering examples. The first row depicts the contact points locally fixed on the climber. The second row shows the two sets of holds for the bouldering examples.*

contact establishment are penalized.

Qualitative results

The figures 4.9 to 4.12 show the comparison of different initial timings with their related optimal timings for both examples. The figure 4.13 and 4.14 show the comparison of optimal timings and ground truth timings for both examples. Each sequence is plotted at 10 fps and each frame has been shifted toward the right, producing an artificial horizontal displacement that were not included in the computed climbing trajectories.

For both examples and for the three presented initializations (slow, fast and medium speed), the timing for the push-off and the in-air phases is well recovered, showing the efficiency of the optimization for ballistic motions. On the contrary, the timing for the first move and for the landing are slower for the ground truth than for the optimal timing. For those phases, either the input motion is not optimal, or our modelization is incomplete or not suitable for the task performed by the climber.

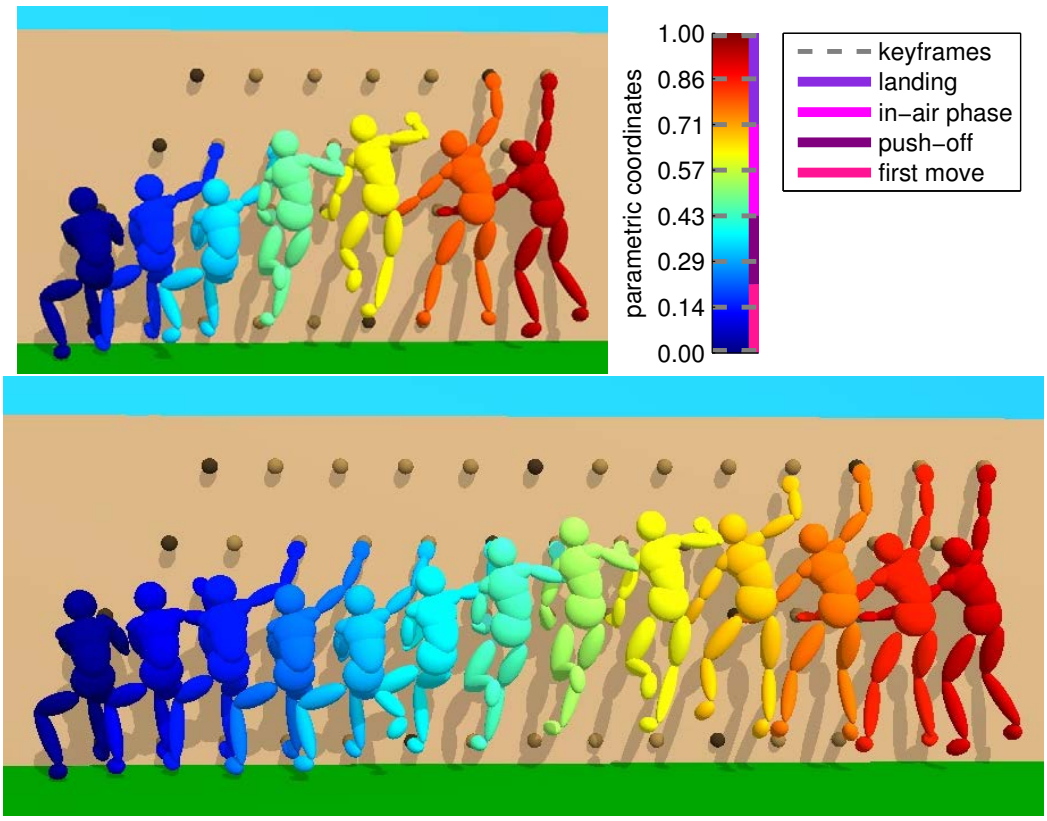


Figure 4.9: *Example 1, top: fast initialization, bottom: optimum. The color bar is the parametric coordinate space. The horizontal axis is the time, at 10 fps.*

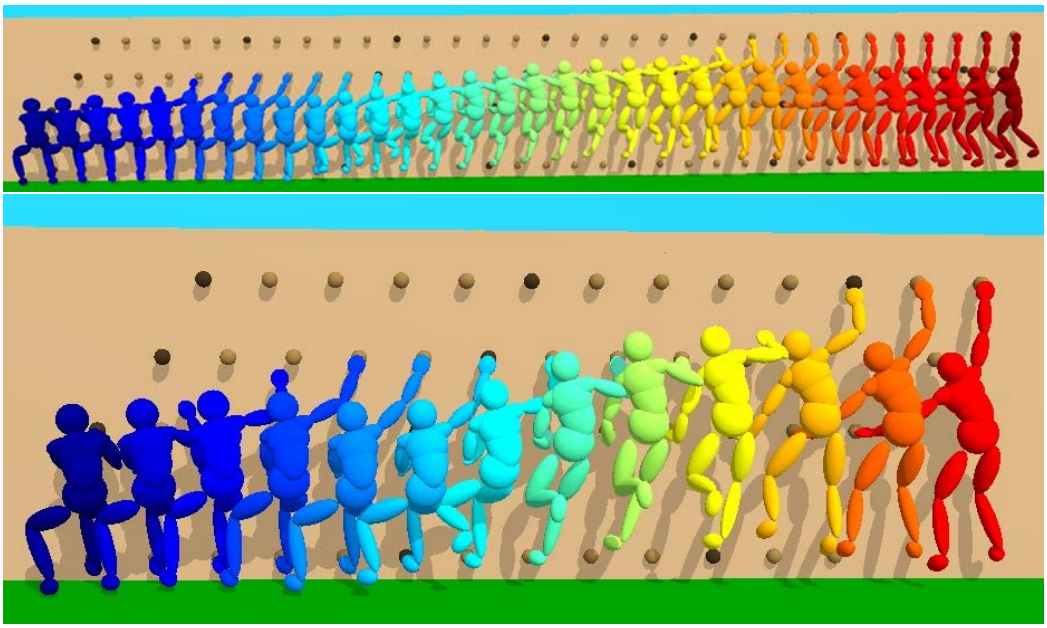


Figure 4.10: *Example 1, top: slow initial timing, bottom: related optimum.*

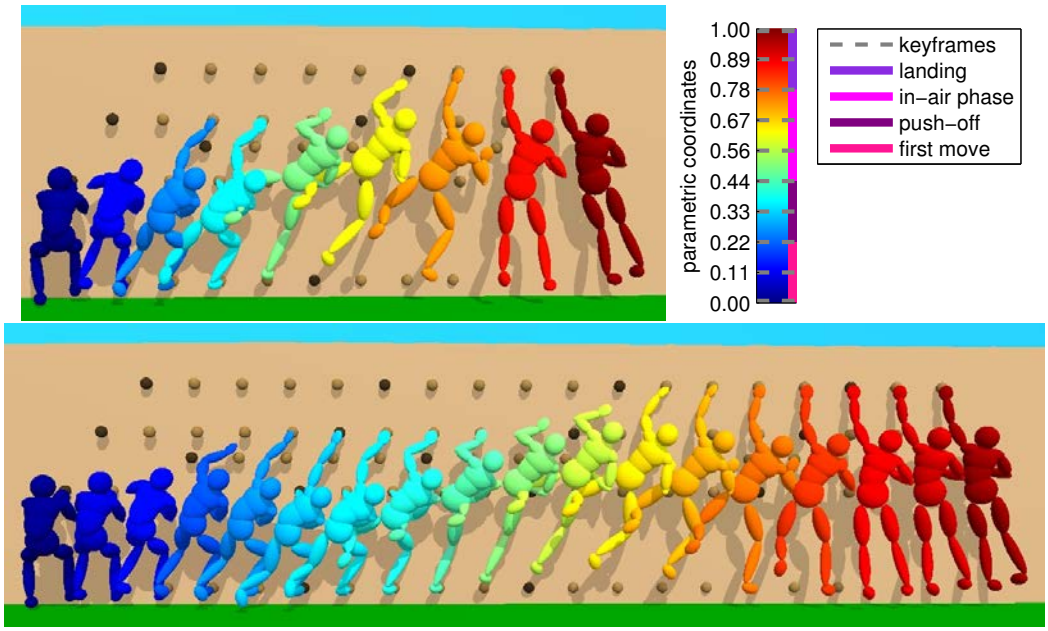


Figure 4.11: *Example 2, an initial fast timing (top left), the optimal solution obtained with this initialization (botton), and the four motion phases in parametric space (top right).*

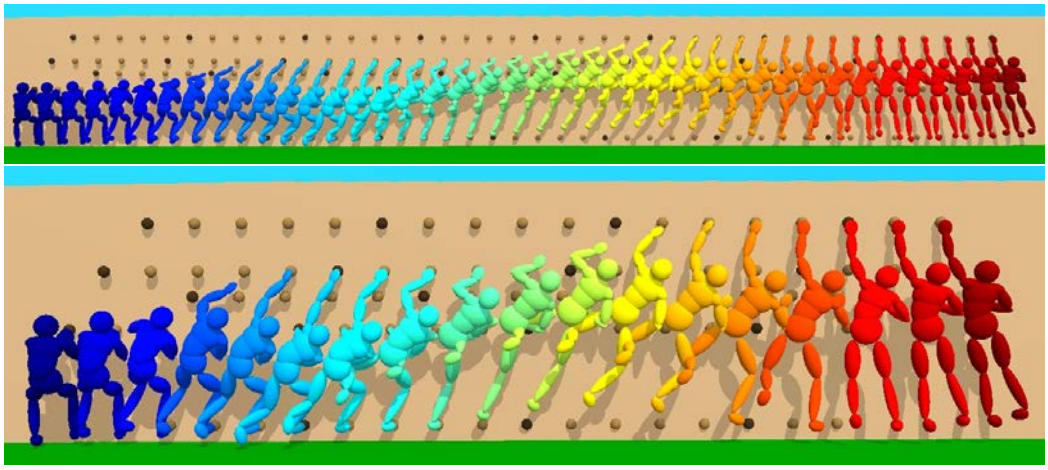


Figure 4.12: *Example 2, slow initialization (top) and corresponding optimal timing (bottom). The time step between two successive hold patterns is 0.1 second. The time step between two dark brown hold patterns is 0.5 second.*

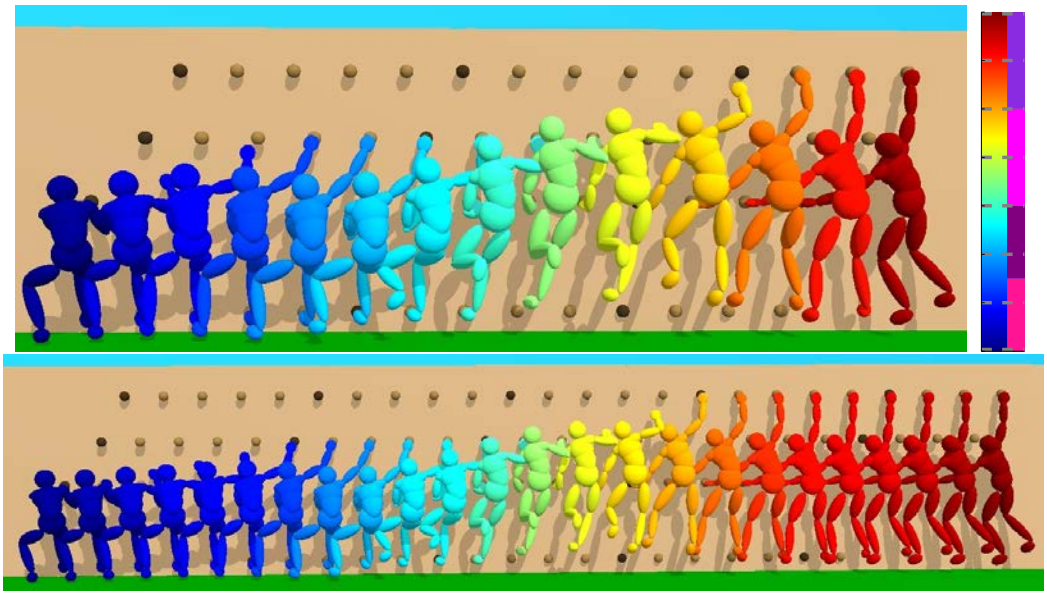


Figure 4.13: *Example 1, optimal timing for a medium-speed initialization (top), approximated ground truth timing (bottom). The ground truth timing is available only at each keyframe.*

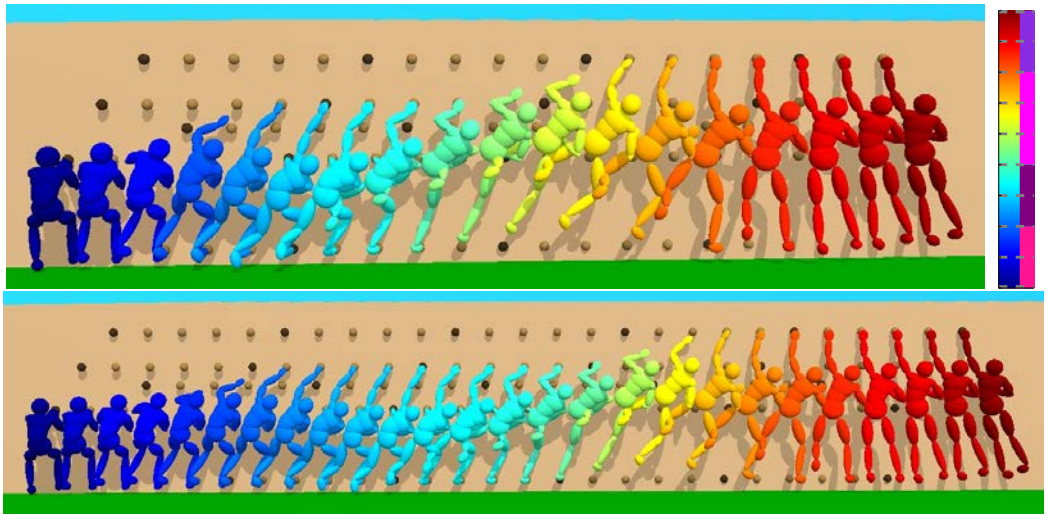


Figure 4.14: *Same as above for the example 2.*

Quantitative results

The optimal solutions for the examples 1 and 2 are presented quantitatively in the figure 4.15. For each graph, four different optimal timings, corresponding to four different initialization are plotted. The different initializations converge to the similar optimal timing, except at the end of the example 1, and at parametric coordinate 0 and 0.33 for the example 2. Moreover, the ground (red dashed line) in quite well recovered between parametric coordinates 0.14 and 0.71 for the first example, and between 0.44 and 0.78 for the second example, which correspond to the ballistic phase of the motion, as mentioned in the qualitative evaluation above.

The cost function values are presented in the table 4.1. For the example 1, the mean total cost is 364 ± 68 at initialization and 234 ± 12 at convergence. The standard deviation reduction is thus of 82.3%. For the example 2, the initial cost is 441 ± 84 and the final cost is 234 ± 1. In this case the standard deviation reduction is of 98.5%. These high percentage of reduction shows that the solutions are close to global minima.

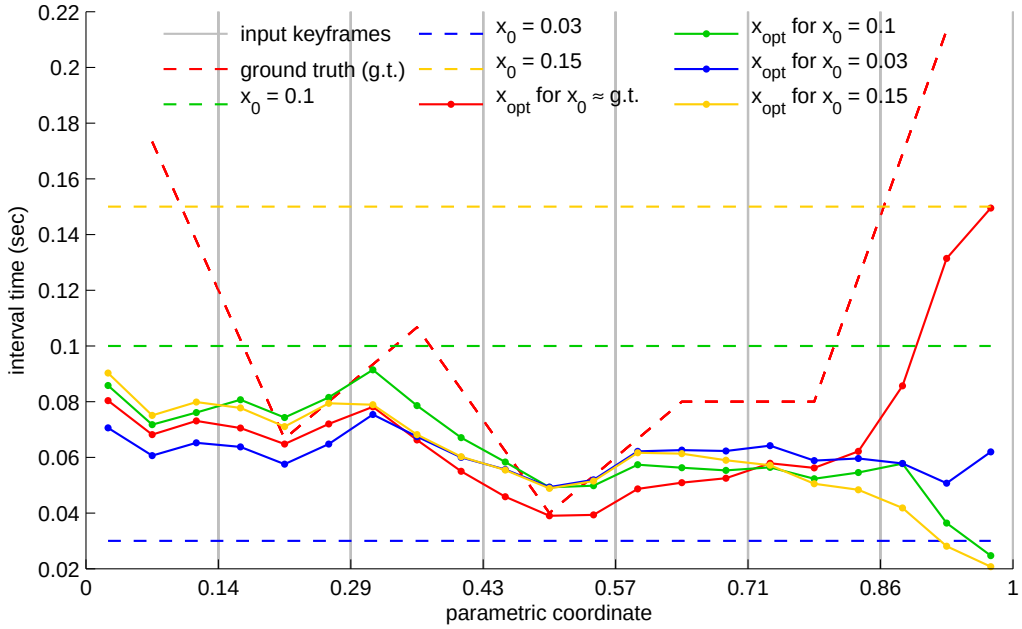
initial timing intervals	initial cost (norm)			optimal cost (norm)		
	actuation (N.m.s)	contact (N.s)	both (∅)	actuation (N.m.s)	contact (N.s)	both (∅)
ground truth	172.3	275.3	324.8	139.8	209.7	252.1
const. 30 ms	270.5	343.6	437.3	125.9	190.9	228.6
const. 100 ms	156.8	245.4	291.2	129.1	191.5	230.9
const. 150 ms	212.2	341.8	402.3	125.6	187.9	226.0

(a) *Example 1, cost function values.*

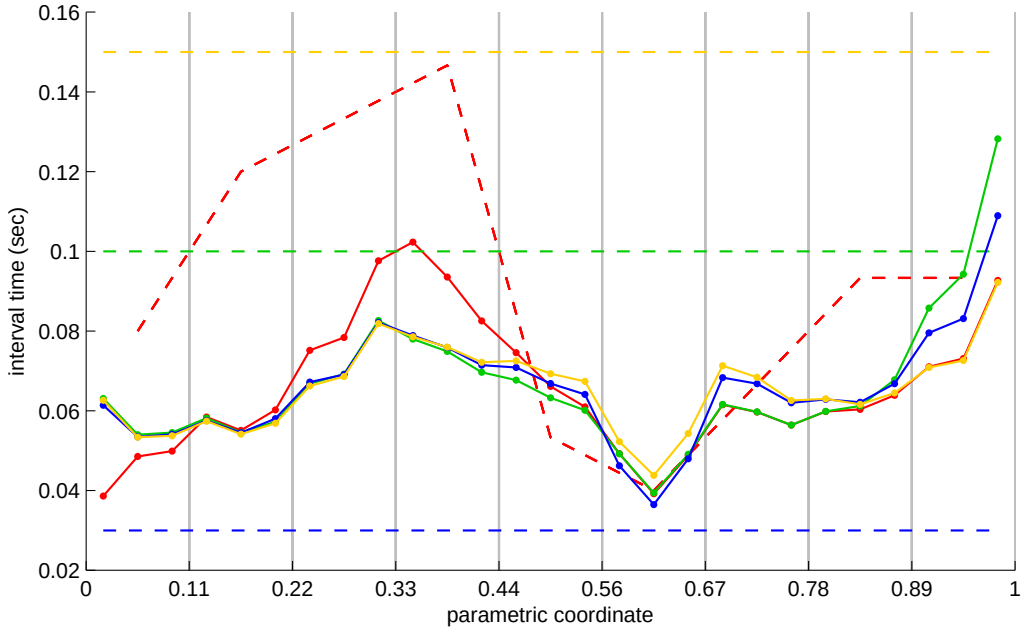
initial timing intervals	initial cost (norm)			optimal cost (norm)		
	actuation (N.m.s)	contact (N.s)	both (∅)	actuation (N.m.s)	contact (N.s)	both (∅)
ground truth	197.9	322.0	378.0	163.3	249.0	297.7
const. 30 ms	328.1	404.3	520.7	160.5	250.3	297.3
const. 100 ms	183.1	309.2	359.3	160.8	251.4	298.4
const. 150 ms	253.2	436.1	504.2	160.8	247.9	295.5

(b) *Example 2, cost function values.*

Table 4.1: *Initial and final cost function values (equation 4.28). The columns "actuation", "contact" and "both" contain respectively the actuation consumption, the air-contact consumption and the total cost (root of squared sum of the first two columns).*



(a) *Example 1, several timing initializations (dashed lines), and their related optimum (continuous lines of the same color).*



(b) *Example 2, same as above.*

Figure 4.15: *Various initializations (x_0's) and corresponding optima (x_{opt}'s) for the examples 1 and 2. The horizontal axis is the parametric coordinate axis. Each point on this axis thus corresponds to a point on the interpolated path. The vertical gray lines are the keyposes. The y-axis are optimization values which are interval times between two successive sub-keyframes, as explained in figure 4.3.*

4.4.2 Input parameters sensitivity and setting

In return for the simple, uncluttered aspect of nonlinear program formulations, the parameters of nonlinear solvers are difficult to set up. In order to demonstrate the difficulty of their adjustment and how we overcome this task, the following subsections deal with the quality of the results with respect to the input settings of the timing optimization solver. We chose to use a gradient-based solver as in [Mordatch 2013], but the Covariance Matrix Adaptation (CMA) Evolution Strategy [Hansen 2006] could also have been used, as in [Al Borno 2012]. The five parameters evaluated in detail in this section are:

1. Finite difference spacing
2. Gradient descent stop criterion
3. Spacing and stop criterion combination
4. Nonlinear solver choice
5. Sampling sensitivity

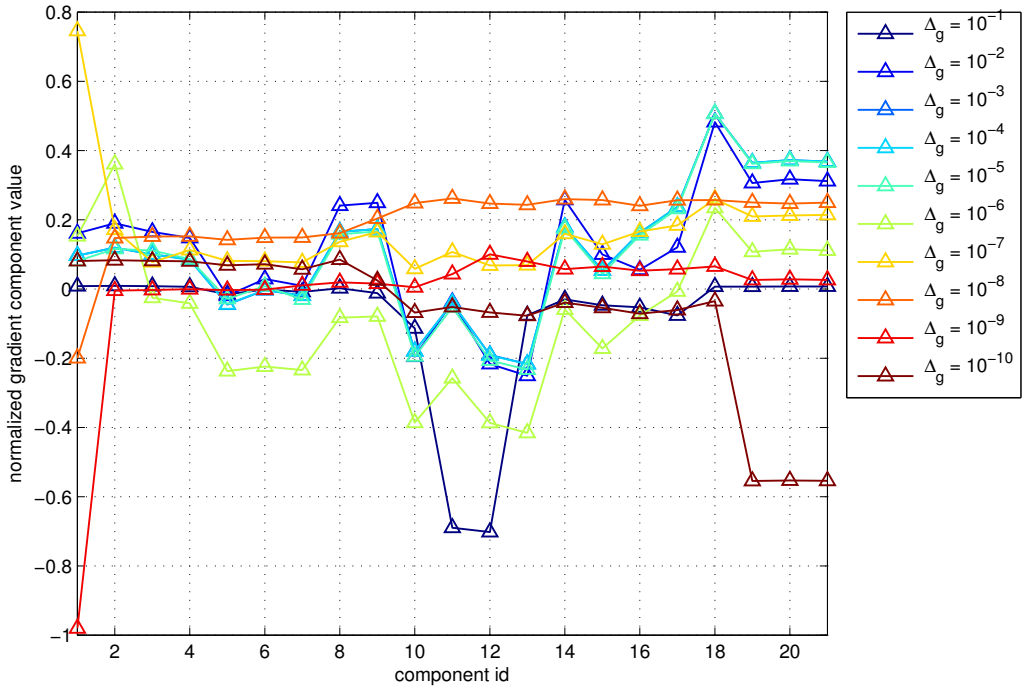
Finite difference spacing

As we use a gradient-descent-based optimization, with a gradient computed by finite difference, the convergence depends mainly on the quality of this gradient. To evaluate this quality, we compute the gradients at several optimization points, for a logarithmic range of finite difference spacings going from $10^{-10}s$ to $10^{-1}s$. These gradients have been normalized for their comparison. We define the confidence range for the finite difference spacing as the set of spacing values leading to similar gradients. For the example 1 (figure 4.16), this confidence range is evaluated to $\left[10^{-5}, 10^{-3}\right]$. For figure 4.17a, this range is $\left[10^{-8}, 10^{-2}\right]$, and for figure 4.17b, it is $\left[10^{-6}, 10^{-4}\right]$. We observe that the confidence range is thinner at the optimum than at the initialization for the second example, which is not the case for the first one, showing the dependence of this range on the observed example. From this study, the value of $\Delta_g = 10^{-5}$ is retained, as a compromise between precision and reliability.

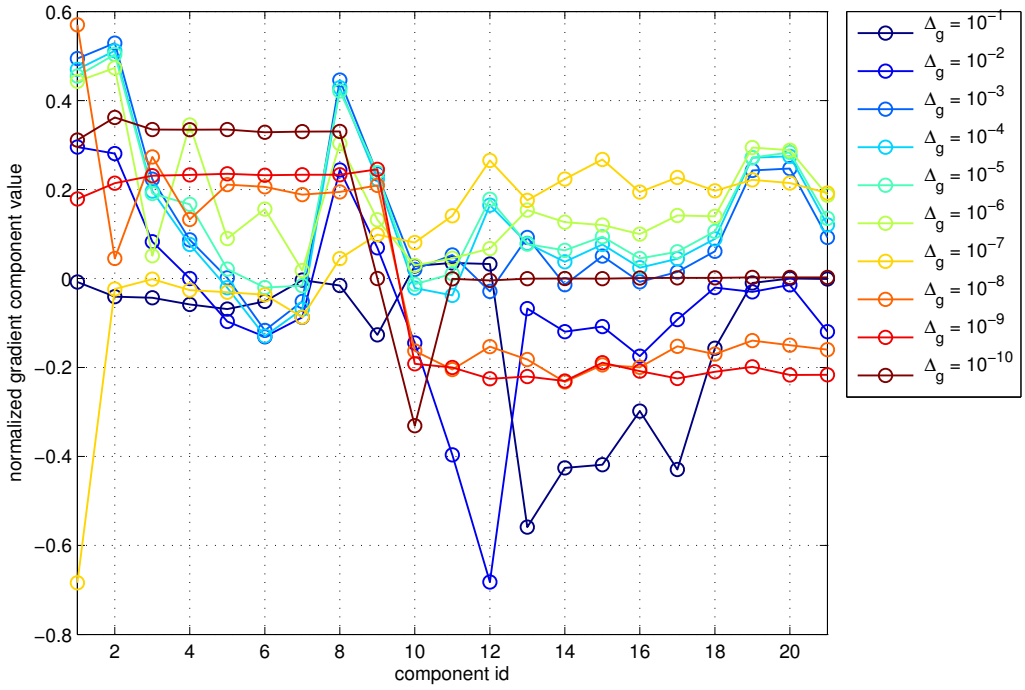
The range of confidence of the finite difference spacing can also be evaluated in a more concise way by using the gradient norms, as in figure 4.18. For norms, the confidence range is defined by the flat horizontal segments. The previous ranges become in this case $\left[10^{-5}, 10^{-3}\right]$, $\left[10^{-8}, 10^{-2}\right]$ and $\left[10^{-6}, 10^{-2}\right]$ respectively. Except for the upper bound of the last range, the ranges of confidence are the same as the ones found with the gradient shapes. The gradient norm is thus a reliable estimator for the confidence range for the gradient spacing.

Gradient descent stop criterion

The stop criterion for the gradient descent is the minimal amount of variation on the unknown x, noted Δ_k. To evaluate the quality of the convergence for a given

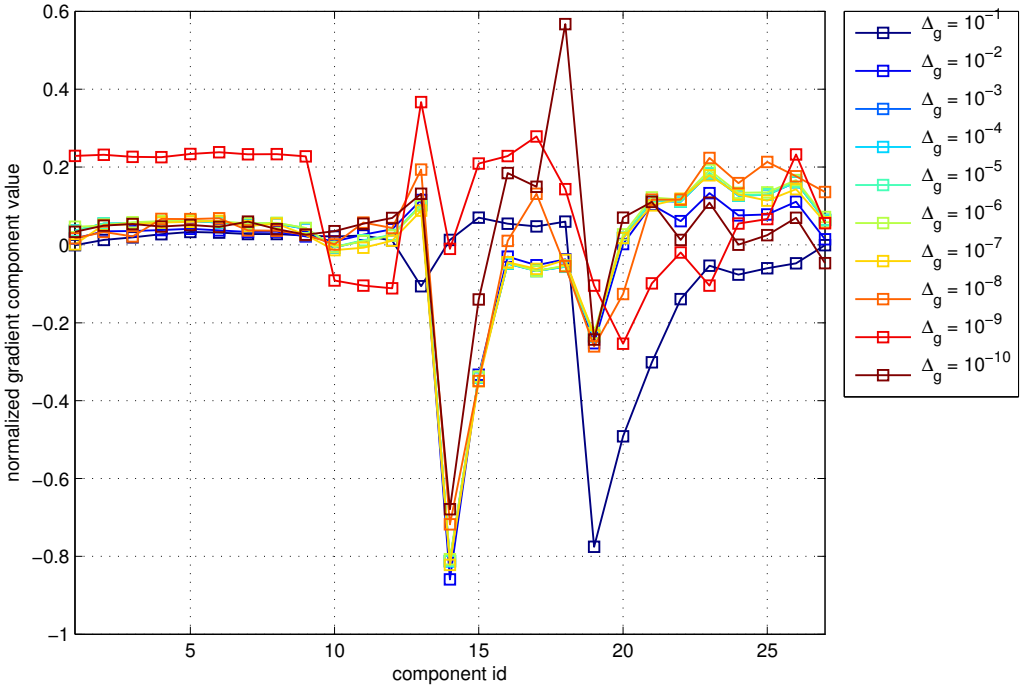


(a) *Gradients at the initial timing, close to the ground truth.*

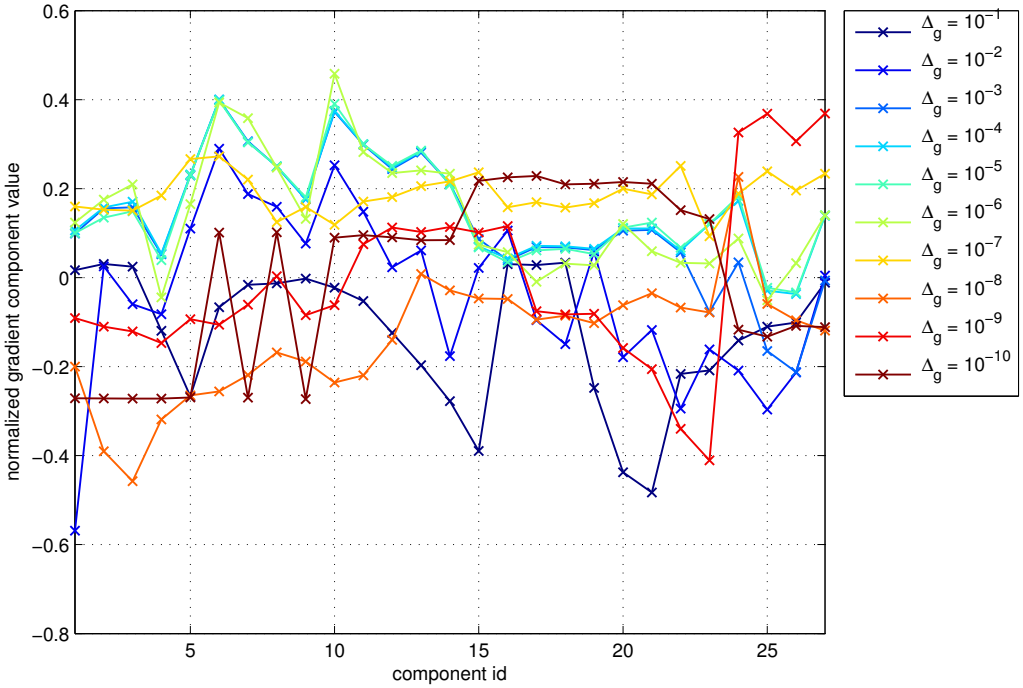


(b) *Gradients at the optimal timing.*

Figure 4.16: *For the example 1, comparison of the component values of normalized gradients of the cost function, for finite difference spacing from* $10^{-10}s$ *to* $10^{-1}s$*, at an initial timing close to the ground truth, and at the corresponding optimal timing. The optimality is obtained with* $\Delta_g = 10^{-6}$*, and with the stop criterion* $\Delta_k = 10^{-4}$*.*



(a) *Gradients close to the ground truth.*



(b) *Gradients at the optimal timing.*

Figure 4.17: *Same as previous page for example 2.*

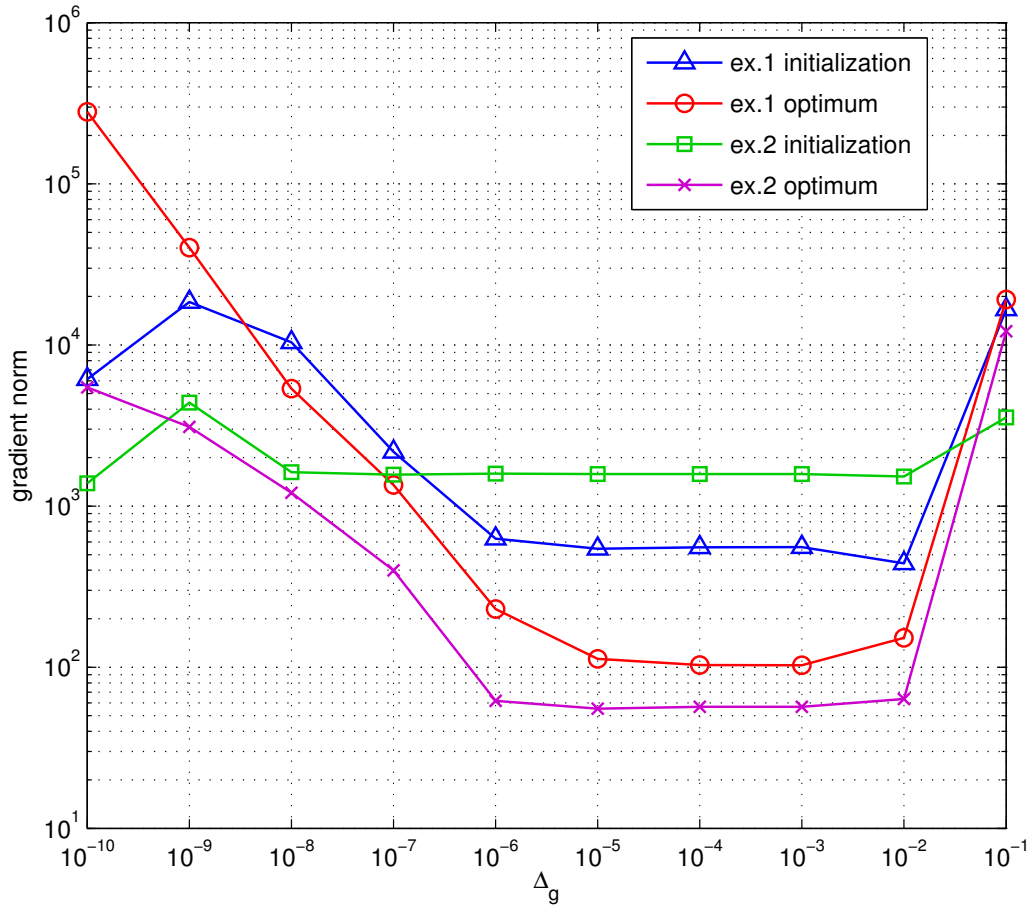


Figure 4.18: *Using gradient norms to summarize figures 4.16 and 4.17.*

Δ_k, we use the nearness of the optimal solutions for the 4 different initializations. Each initialization is a vector of size N_x, noted x_0. The four different x_0's used are constant vectors of 0.03s, 0.1s and 0.15s, and a timing close to the ground truth timing.

The figure 4.19 shows this evaluation for the example 1. For this example, $\Delta_k \leq 10^{-8}$ makes the program to stay at the initialization, leading to very different timing optima. On the contrary, the solutions with $\Delta_k = 10^{-5}$ are the most similar. The figure 4.20 presents the same study for the example 2. For this case the best convergence is also obtained for $\Delta_k = 10^{-5}$.

Note that the optimal timing differences between parametric coordinates 0.86 and 1.0 in the first example are not as obvious in the second example. Moreover for the second example, $\Delta_k \leq 10^{-8}$ gives optimal timings that are not the initial values, except for $x_0 = 0.03$, which is not the case in the example 1. Thus the convergence behavior depends also on the input motion.

To be able to process more systematically the convergence quality, we compute the standard deviations of the optimal timings. By this way, most of the information

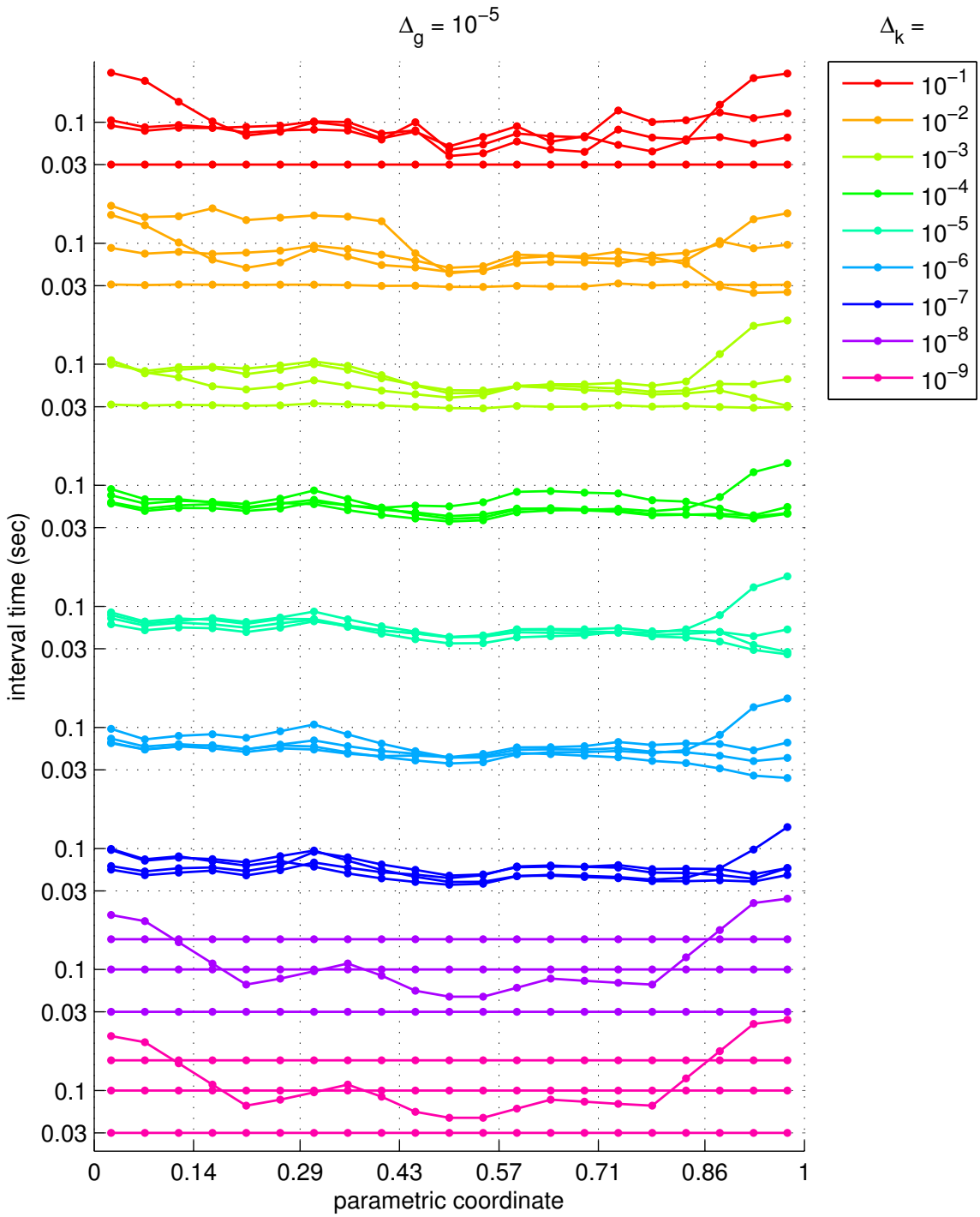


Figure 4.19: *Example 1, convergence variations for stop criteria (Δ_k) from $10^{-9}s$ to $10^{-1}s$. The finite difference spacing is $\Delta_g = 10^{-5}$. Each color represents 4 solutions obtained with 4 different initializations.*

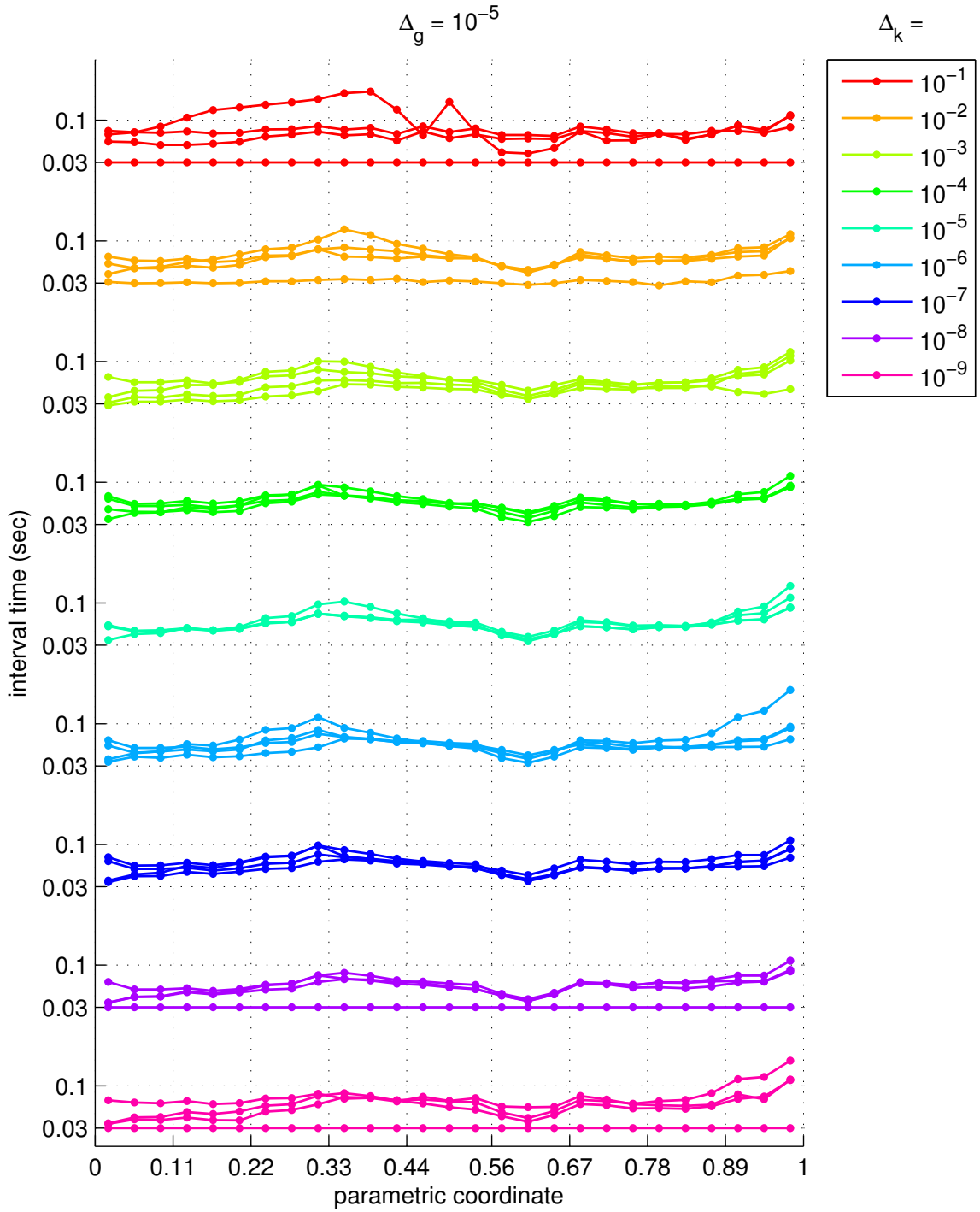
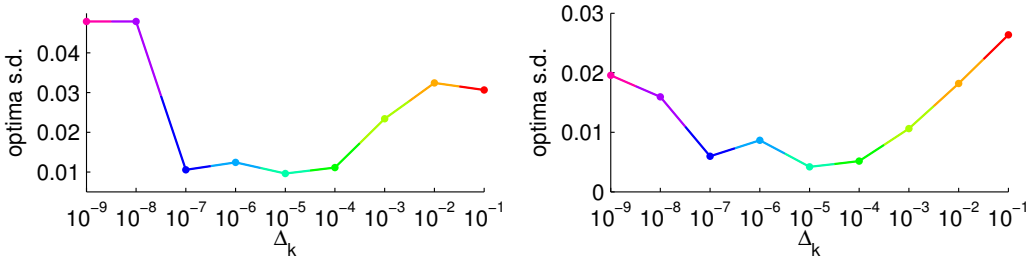
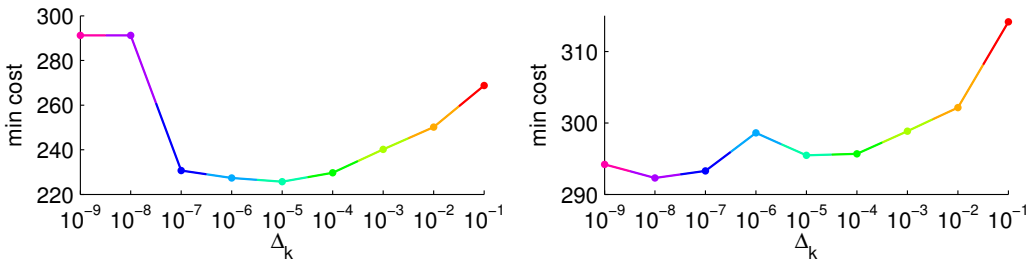


Figure 4.20: *Same study as in the previous page for the example 2.*



(a) *Example 1, standard deviations.* **(b)** *Example 2, standard deviations.*

Figure 4.21: *Convergence indicator 1: Standard deviations of the four optimal solutions for each Δ_k values.*



(a) *Example 1, minimal cost.* **(b)** *Example 2, minimal cost.*

Figure 4.22: *Convergence indicator 2: The minimal costs among the 4 optimal solutions for each Δ_k.*

of figures 4.19 and 4.20 can be gathered in 1-D values, as shown in the figure 4.21. Along with this first indicator of the convergence quality, a second indicator can be used, which is the minimal cost among the four optimal costs corresponding to the different timings. The figure 4.22 presents these costs for the studied examples. By combining these four graphs, we can confirm the choice of $\Delta_k = 10^{-5}$.

The x_0 having lead to the cost minima of figure 4.22 are shown in the 2 tables of figure 4.23. We conclude from these tables that there is no best initialization, and that the initialization close to the ground truth (g.t.) surprisingly gives the best cost only in 11% of the studied cases. The ground truth is thus not always the best initialization.

Spacing and stop criterion combination

Although the standard deviation and the minimal cost give less information than the direct observation of the optimal timings, their dimension of 1 allows us to evaluate the inter-dependency of the two input parameters Δ_g and Δ_k in 2-D graphs, as presented in the figures 4.24 and 4.25. In these graphs, the confidence region for the parameters Δ_g and Δ_k is defined as the dark blue area. The white dashed lines

Δ_k	10^{-9}	10^{-8}	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
min cost x_0	0.1	0.1	0.03	0.03	0.15	0.1	0.15	0.1	0.1

(a) *Example 1, min cost / x_0 correspondence.*

Δ_k	10^{-9}	10^{-8}	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
min cost x_0	g.t.	0.15	0.15	0.1	0.15	0.15	g.t.	0.15	0.15

(b) *Example 2, min cost / x_0 correspondence.*

Figure 4.23: *The initial timings that give the minimal costs of figure 4.22.*

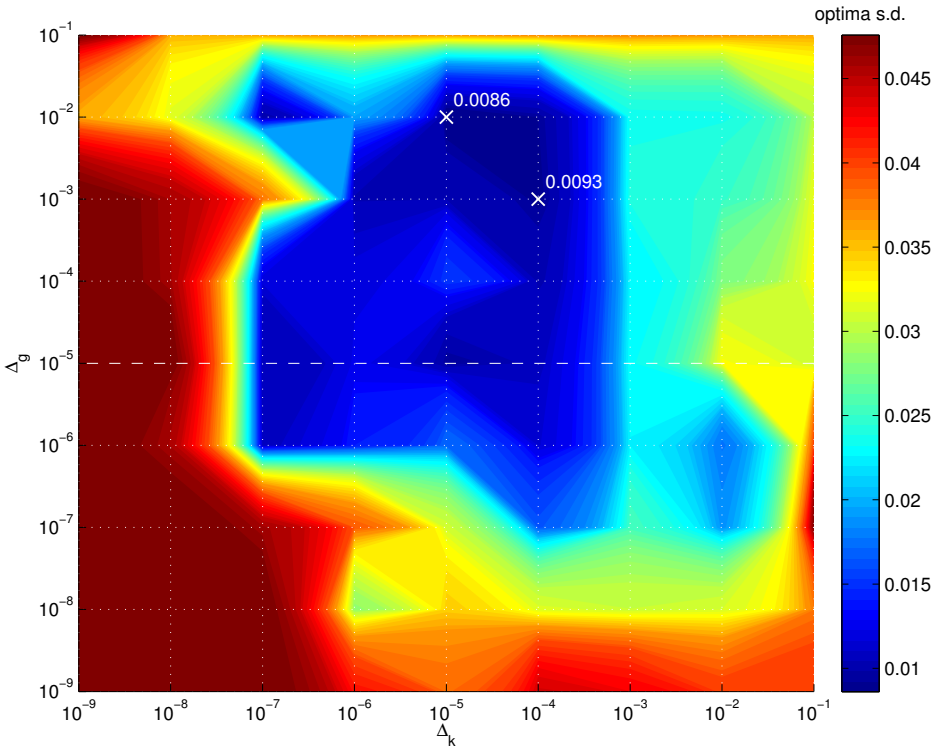
$\Delta_g \backslash \Delta_k$	10^{-9}	10^{-8}	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
10^{-1}	0.1	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
10^{-2}	0.15	0.15	0.03	0.15	0.1	0.03	0.15	0.1	0.15
10^{-3}	0.1	g.t.	0.03	0.03	0.1	0.03	0.1	0.1	0.1
10^{-4}	0.1	g.t.	0.03	0.03	0.1	0.03	0.1	0.1	0.1
10^{-5}	0.1	0.1	0.03	0.03	0.15	0.1	0.15	0.1	0.1
10^{-6}	0.1	g.t.	0.03	0.03	0.1	0.03	0.1	0.03	0.1
10^{-7}	0.1	0.1	0.1	0.03	0.15	0.1	0.15	0.03	0.1
10^{-8}	0.1	0.1	0.1	0.15	0.1	0.1	0.1	0.1	0.1
10^{-9}	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1

(a) *Example 1, min cost / x_0 correspondence for the figure 4.24b.*

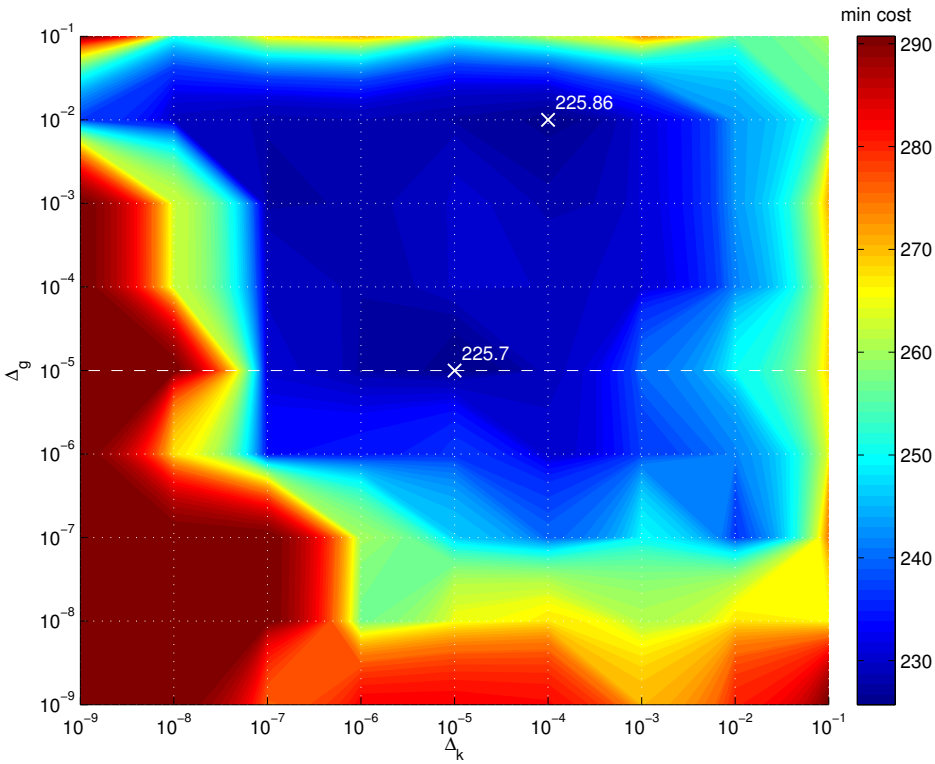
$\Delta_g \backslash \Delta_k$	10^{-9}	10^{-8}	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
10^{-1}	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
10^{-2}	g.t.	0.15	0.15	0.1	0.03	0.15	0.15	0.1	0.15
10^{-3}	0.15	g.t.	0.15	0.1	g.t.	0.1	g.t.	0.15	0.15
10^{-4}	g.t.	g.t.	0.15	0.1	0.03	0.15	0.15	0.15	0.15
10^{-5}	g.t.	0.15	0.15	0.1	0.15	0.15	g.t.	0.15	0.15
10^{-6}	0.03	0.03	0.15	0.15	0.15	0.03	g.t.	0.1	0.15
10^{-7}	g.t.	g.t.	g.t.	g.t.	g.t.	g.t.	g.t.	g.t.	0.1
10^{-8}	g.t.	g.t.	g.t.	0.03	0.15	0.03	0.03	0.03	0.03
10^{-9}	0.1	0.1	0.1	0.1	0.1	g.t.	0.1	g.t.	0.1

(b) *Example 2, min cost / x_0 correspondence for the figure 4.25b.*

Table 4.2: *The initializations that gave the best optimal timings, chosen among the four candidates $x_0 \approx$ ground truth (g.t.), $x_0 = 0.03s$, $x_0 = 0.1s$ and $x_0 = 0.15s$.*

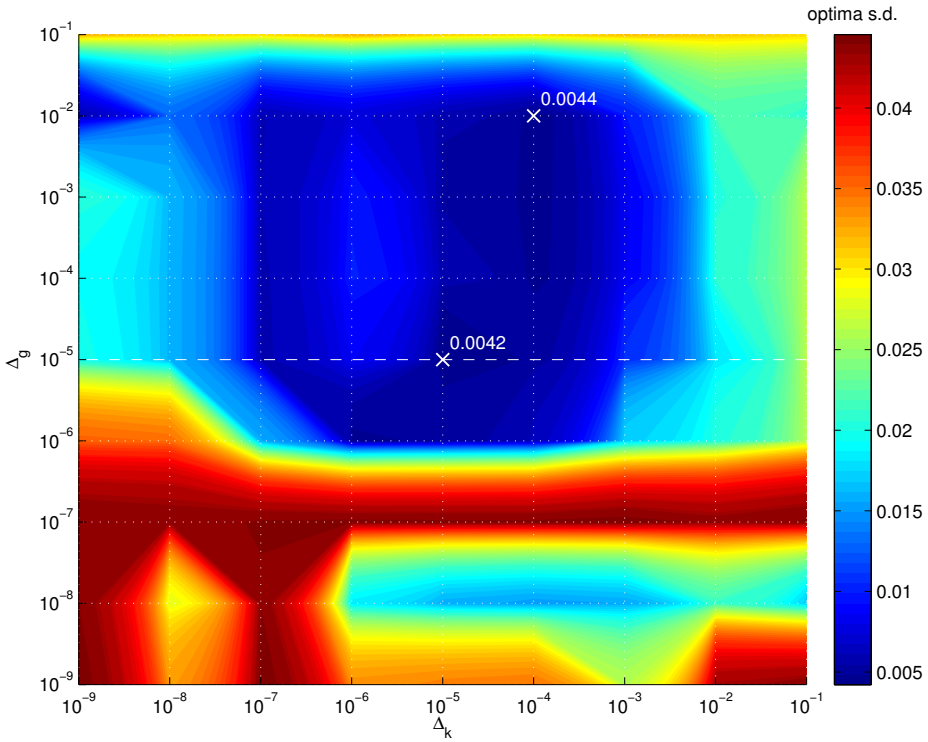


(a) *Example 1, standard deviations of optimal timings for four different initializations, in the optimizer parameter space* Δ_g–Δ_k.

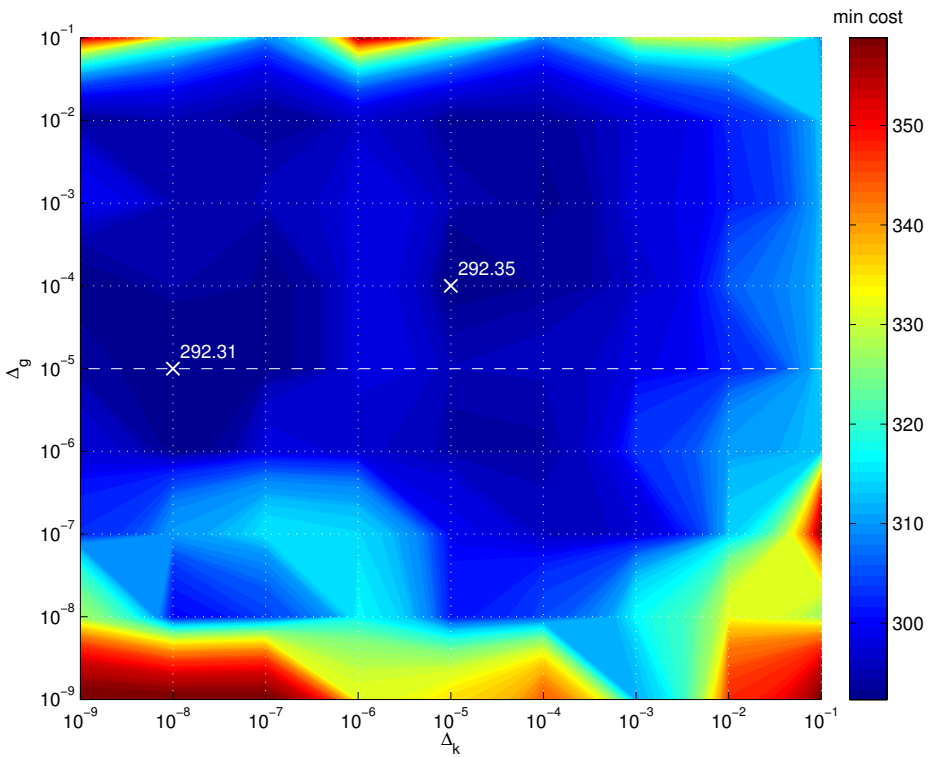


(b) *Example 1, optimal cost in the same space as above. The initializations corresponding to the retained optima are shown table 4.2a.*

Figure 4.24: *Example 1, convergence evaluation for the parameters* Δ_g *and* Δ_k.



(a) *Example 2, standard deviations.*



(b) *Example 2, optimal costs. Corresponding initializations are printed table 4.2b.*

Figure 4.25: *Example 2, same as previous page.*

correspond to the curves of figures 4.21 and 4.22. The two white numbers inside each graph are the two smallest values in the graph.

The figures 4.24 and 4.25 show that the standard deviation is not convex in the space Δ_g–Δ_k. Therefore, a gradient-descent-based optimization in this space would not success, unless starting close to the solution. This justifies the brute force evaluation we used here.

From those four plots, we confirm the choice of $\Delta_g = 10^{-5}$, because the two cost minima and one minimal standard deviation are obtained with this value, and the choice of $\Delta_k = 10^{-5}$, because it has led to the 2 minima for the standard deviation, 1 minimal cost and one second minimal cost. Actually those graphs are the ones which have been used for bootstrapping the two previous parameter evaluations.

The table 4.2 presents the best initializations corresponding to the figures 4.24b and 4.25b. By observing the dark blue area in the two tables, we confirm that no initialization is always the best. In this wider study, the ground truth initialization is the best for 15.4% of the studied cases. This percentage is 16.0% for $x_0 = 0.03s$, 36.4% for $x_0 = 0.1s$ and 32.1% for $x_0 = 0.15s$. $x_0 = 0.1s$ has thus a tendency to give better results.

Nonlinear solver choice

Two solvers have been tested, each of them with its own algorithm. The first solver is the function `dtrnlspbc` from the Intel Math Kernel Library (MKL). This solver uses the Trust-Region algorithms [Conn 2000] to solve nonlinear least square problems with linear bound constraints. The second solver is our own implementation of a gradient descent algorithm, with an exponential step size adaptation. This algorithm has been designed to be easily implementable. For more technically advanced adaptive gradient descent algorithms, one could look for instance at the work by Hazan et al. [Hazan 2007]. The algorithm 1 presents in pseudo-code the adaptive gradient descent method we propose. The idea behind the algorithm is to increase or decrease the step size as far as possible on an exponential scale, after each gradient computation.

The MKL solver has 5 different stop criteria, corresponding to the minimal precision on the x variation (Δ_k in our case), on the trust region size, on the cost variation, on the cost value and on the norm of its gradient. To allow a comparison with our gradient descent having only a single stop criterion, all the parameters of MKL are set to Δ_k. Note that most of the MKL parameters are not used in our case solely to preserve the simplicity of the algorithm. If a more robust algorithm is needed, those criteria can be directly added by taking the variable *update* as example.

The results obtained with MKL are shown in figures 4.26 and 4.27. Similarly to the previous results, Trust-Region results show non-convex spaces, as we can see at the bottom of figures 4.26b, 4.27a and 4.27b, as well as with the blue nuances of the confidence region figure 4.26a. As opposed to the previous graphs, the confidence region has no lower bound on Δ_k (the horizontal axis), even after having increased

Inputs : $\mathtt{F} : \mathbb{R}^n \to \mathbb{R}$ (cost function), $x_0 \in \mathbb{R}^n$ (initialization), $\Delta_g \in \mathbb{R}$ (gradient computation spacing), $\Delta_k \in \mathbb{R}$ (step unit, stop criterion)
Output: $x \in \mathbb{R}^n$ (local minimum of $\mathtt{F}$)

```
x ← x_0
k ← 0
repeat
    G ← normalized gradient of F at x on spacing Δ_g
    S(k) := x − 2^k Δ_k G          // exponential step function
    if F(S(k)) < F(x)
        while F(S(k+1)) < F(S(k))
            k ← k+1                // increase step size
    else
        while F(S(k)) > F(S(k−1))
            k ← k−1                // decrease step size
    update ← ||x − S(k)||
    x ← S(k)
until update < Δ_k
```

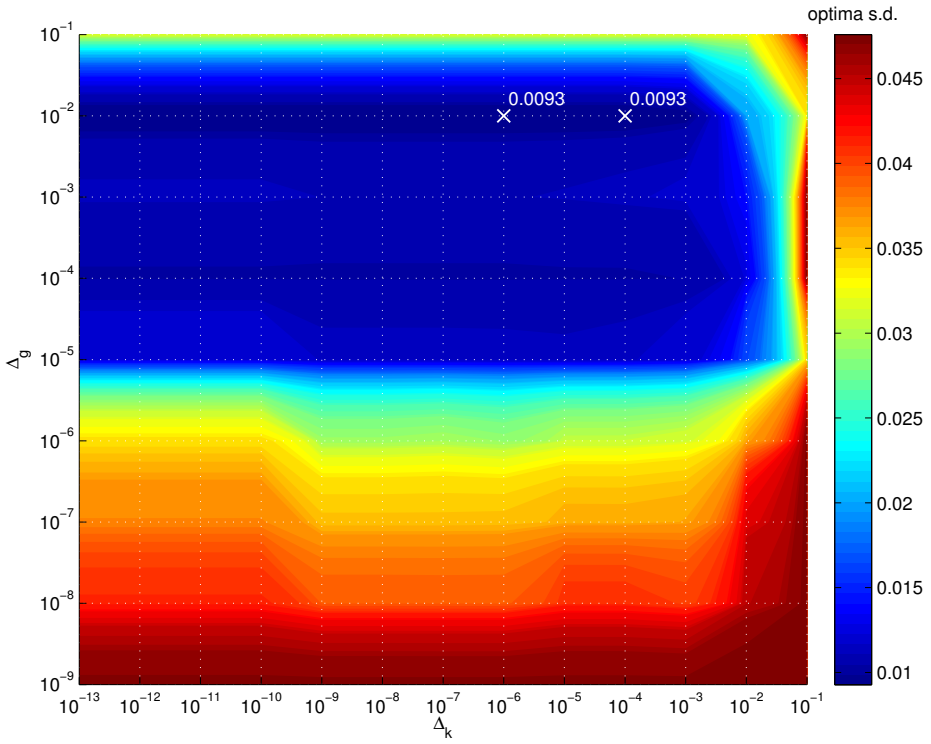
Algorithm 1: Exponentially adaptive gradient descent

	global min s.d. (ms)		global min cost	
	example 1	example 2	example 1	example 2
Ours	**8.6**	4.2	**226**	**292**
MKL	9.3	**3.4**	230	300

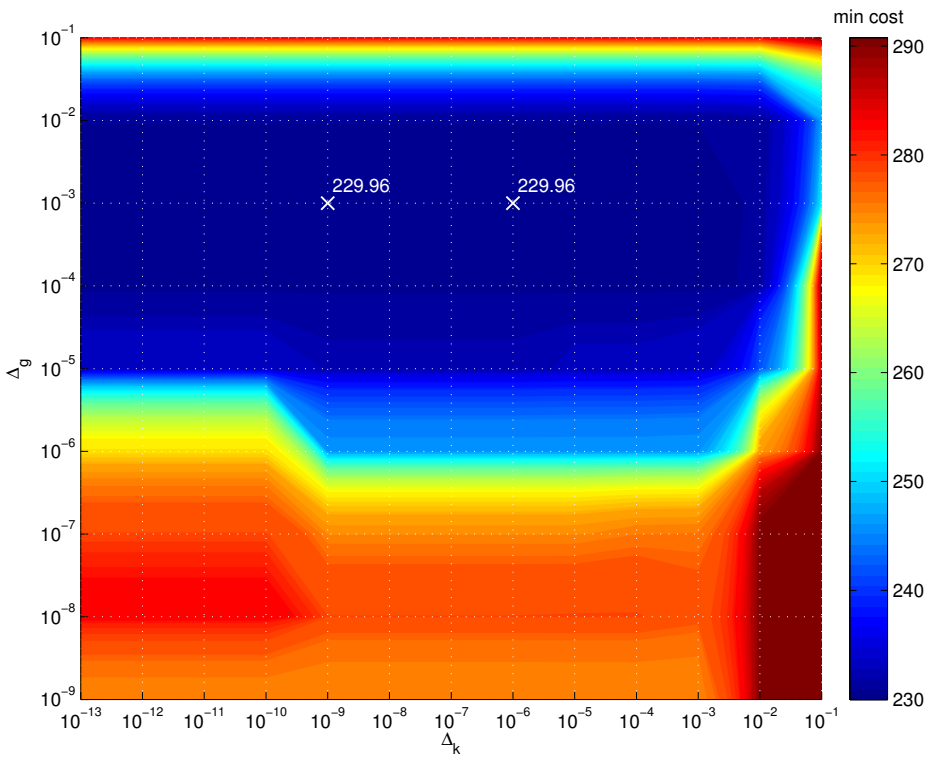
Table 4.3: *Numerical comparison of our algorithm with MKL.*

the range of evaluation up to the lower bound $\Delta_k = 10^{-13}$. The explanation is that the Trust-Region algorithm evaluates the step size for the gradient descent independently of Δ_k unlike the previous algorithm, but based on the current Trust-Region size. We suspect that similar results would be obtained with our algorithm by initializing k (in the algorithm 1) to a higher number, such as 50 for instance.

The table 4.3 gathers the minima obtained with both algorithms. Our algorithm gives the best minimal cost for both examples, and the standard deviation is better in our case for the example 1. Therefore we chose to use our algorithm instead of MKL for the timing optimization. Furthermore, in our case no license is needed, which facilitates the parallelization on a cluster of computers for instance.

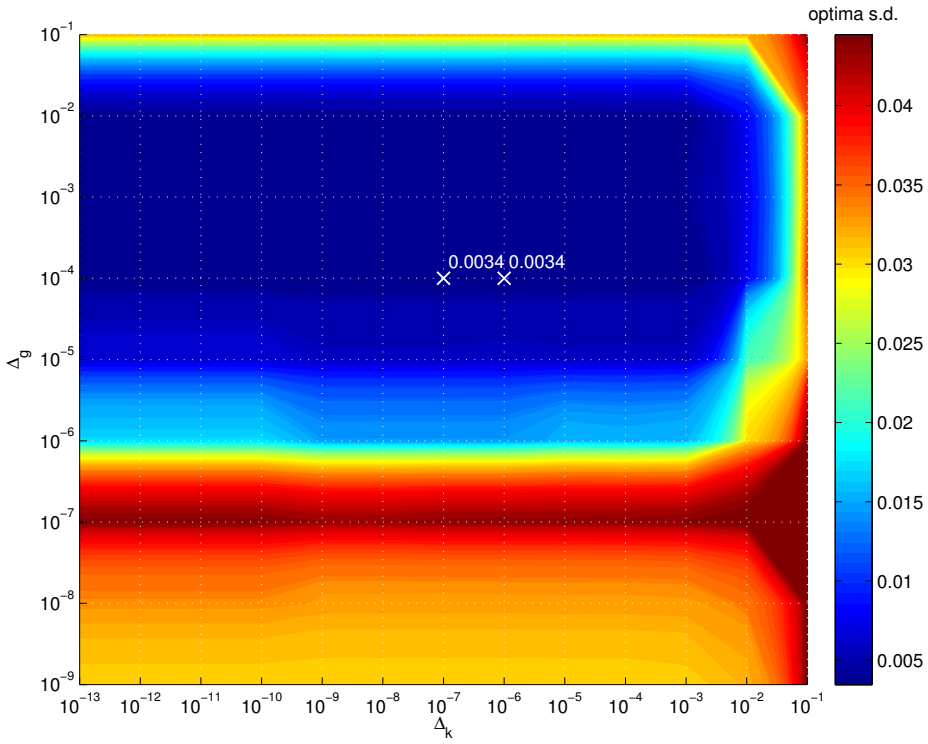


(a) *Example 1, Trust-Region standard deviations.*

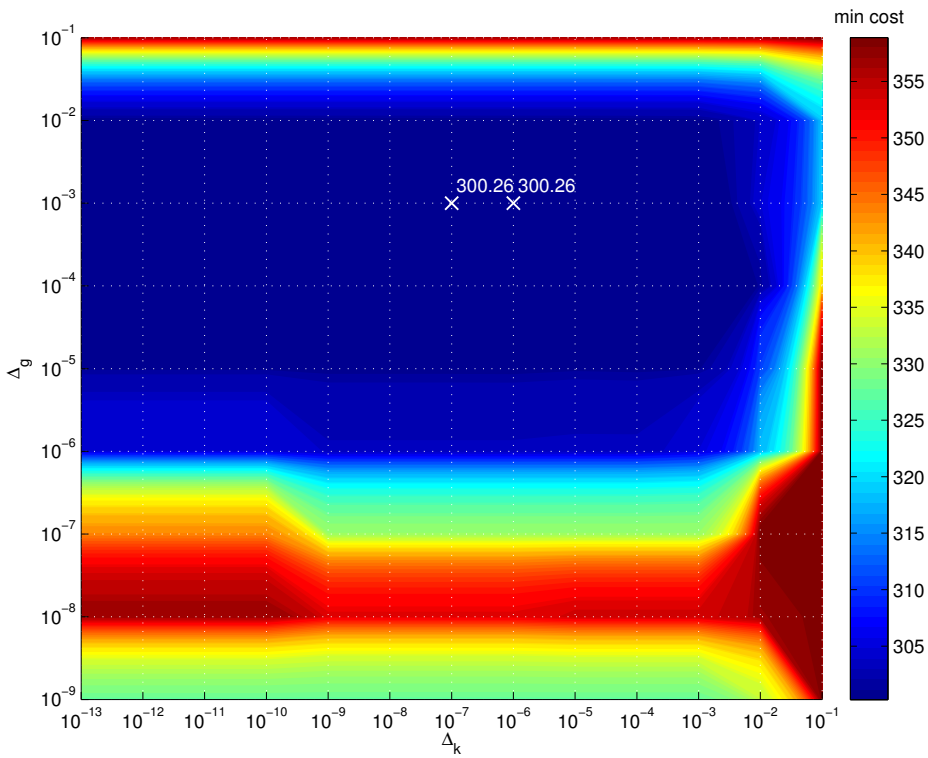


(b) *Example 1, Trust-Region optimal costs.*

Figure 4.26: *Example 1, same graphs as in figure 4.24 computed with a Trust-Region algorithm, instead of the algorithm 1.*



(a) *Example 2, Trust-Region standard deviations.*



(b) *Example 2, Trust-Region optimal costs.*

Figure 4.27: *Example 2, Trust-Region results corresponding to the figure 4.25.*

Fig.	Description	
4.28	Optimal timings	for each plot, N_{id} is varying, $N_x = constant$.
4.29		for each plot, $N_{id} = constant$, N_x is varying.
4.30		$N_{id} \simeq N_x$, $N_{id} \simeq 2N_x$ and $N_{id} \simeq 3N_x$.
4.31	Optimal costs.	
4.32	Computation time.	
4.33	Min x-component value checking. (lower bound = $1ms$)	
4.34	Number of cost evaluations checking. (max = 10^5 calls)	

Table 4.4: *Sampling study, figure list.*

Sampling sensitivity

In this subsection we study the convergence behavior with respect to the 2 sampling parameters, which are the number of frames used for the inverse dynamics computation N_{id}, and the number of optimization variables N_x. For this study, the example 1 is solved using the algorithm 1, with the parameters $\Delta_g = 10^{-5}s$, $\Delta_k = 10^{-5}s$ and $x_0 = 0.1s$, and the aim is to find similar timings for different pairs of N_{id}–N_x. The table 4.4 provides the list of the different plots, and the table 4.5 gives the associated conclusions.

The N_{id}–N_x space is sampled with a 10-by-10 grid, with the N_{id} range going from 20 to 200, and the N_x range going from 14 to 140. The size of the grid has been chosen to be less than the number of cores we used for the evaluation. The computation has been done on 116 Intel cores, with speeds from 2.4 GHz to 3.3 GHz. The total CPU time (or user time) is 19 days and 12 hours. The real time is about 1 night. The details of the computation time can be found in the figure 4.32. The lower bound for N_x has been chosen to be twice the number of keyframe intervals, which is 7 for the example 1. The lower bound for N_{id} has been chosen to be approximately three times this number. The upper bounds have been chosen as ten times the lower bounds, according to the maximal time of computation that we wanted to be around 12 hours.

Interestingly, some noise in the optimal timing is observed when $N_{id} \leq N_x$ (see

Fig.	Conclusions
4.28	No correlation between optima similarities and N_x variations.
4.29	Slight correlation between optima similarities and N_{id} variations.
4.30	No obvious similarity correlations. Timing noise occurs for $N_{id} \simeq N_x$.
4.31	The cost function depends strongly on N_{id}.
4.32	The computation time is less than 1h iff $N_{id} + N_x < 130$.
4.33	The x lower bound is reached mostly for $N_{id} < N_x$.
4.34	The algorithm converges in most of the cases.

Table 4.5: *Sampling study, per figure conclusions.*

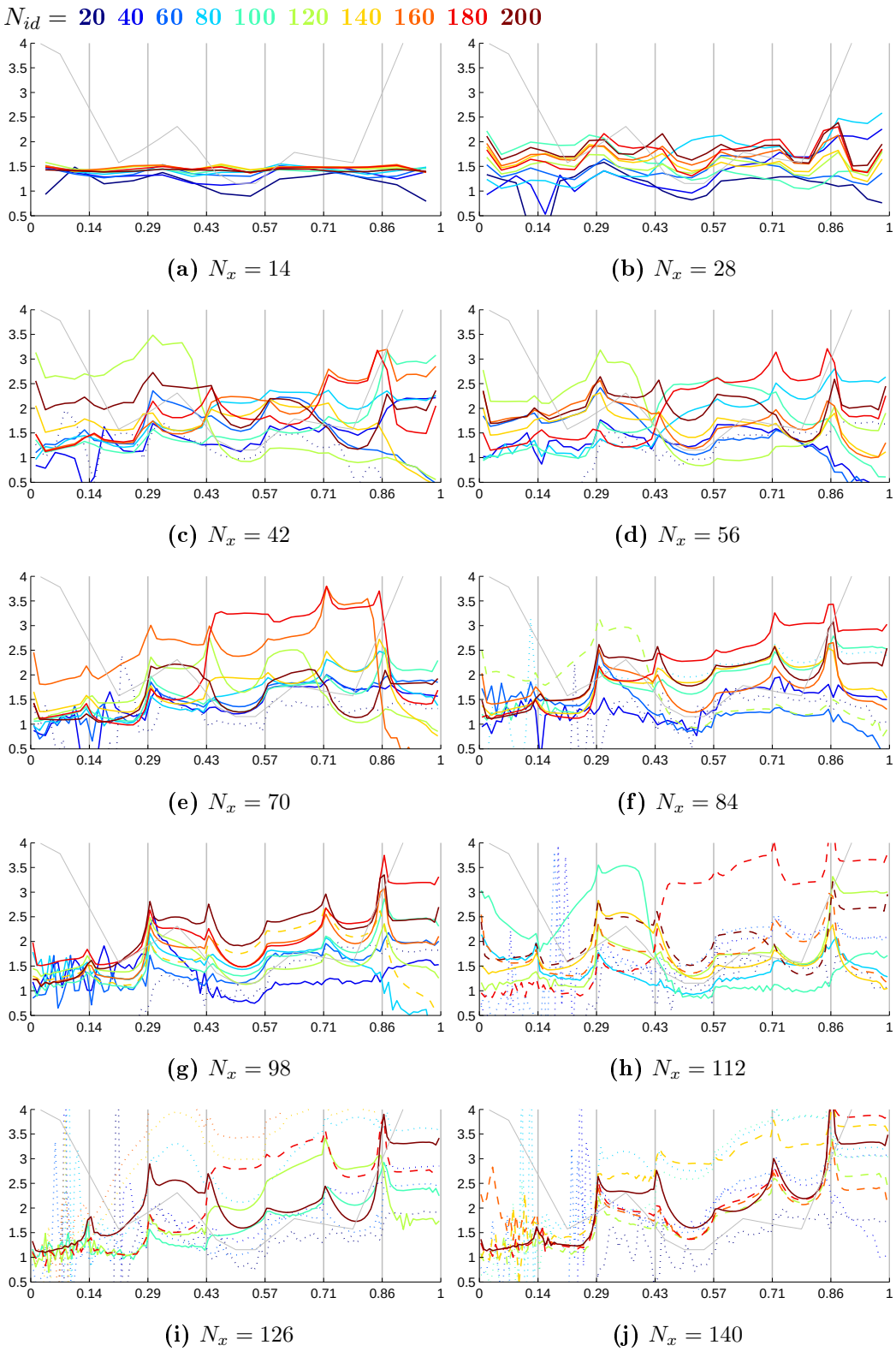
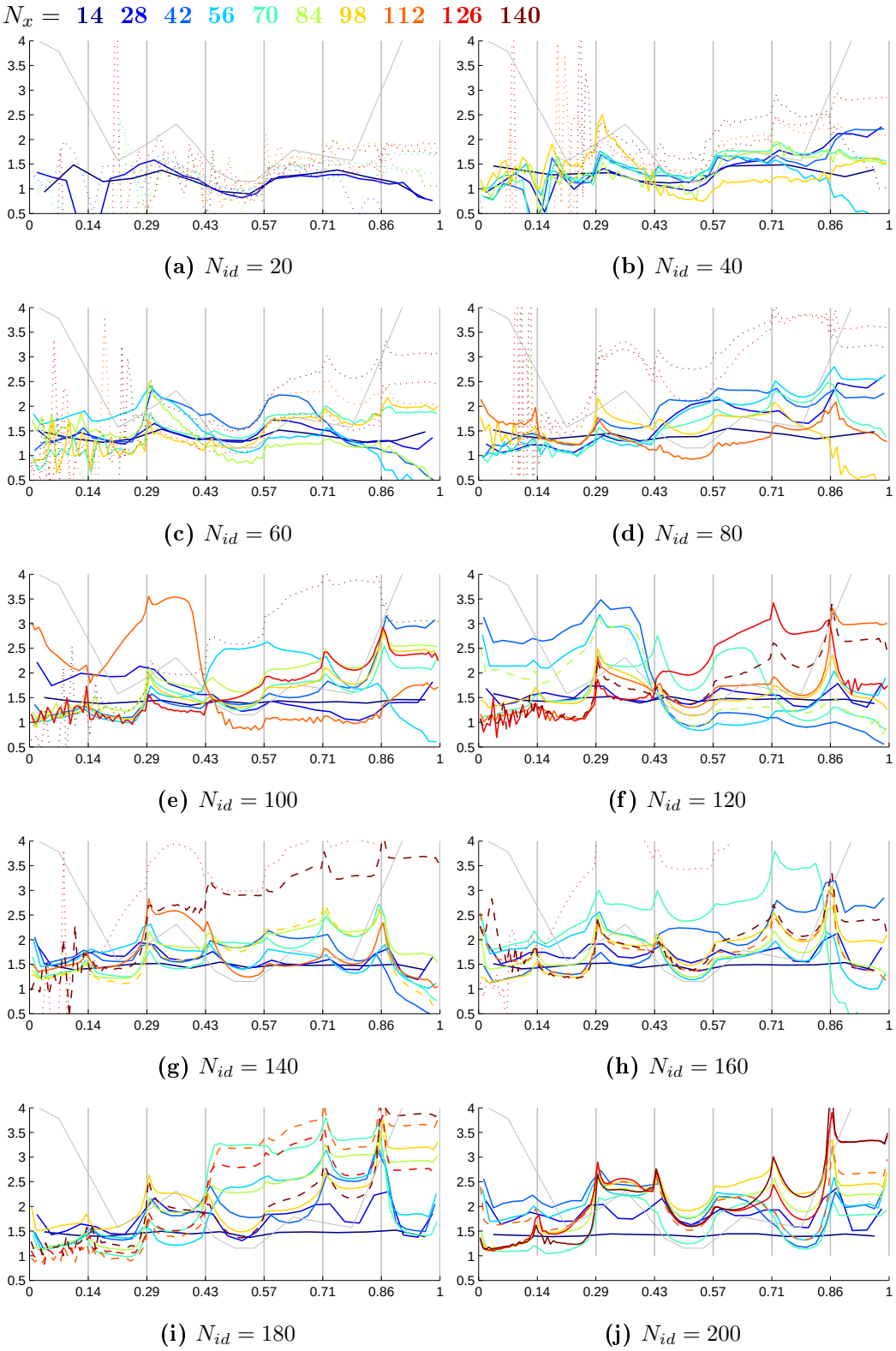
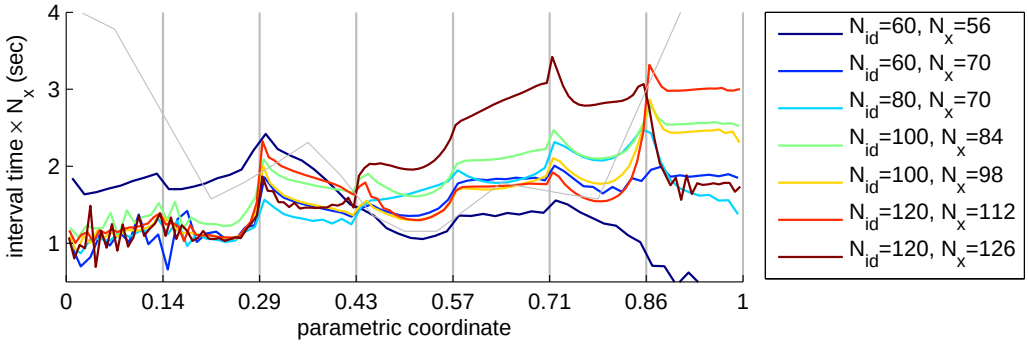


Figure 4.28: *Sampling study, comparison of optimal timings for several N_{id} values, at N_x constant. The horizontal axes are the parametric coordinates. The vertical axes are the sub-keyframe time intervals times the total number of intervals (N_x). Dashed lines: maximum number of cost function evaluations reached. Dotted lines: x lower bound reached (1ms).*

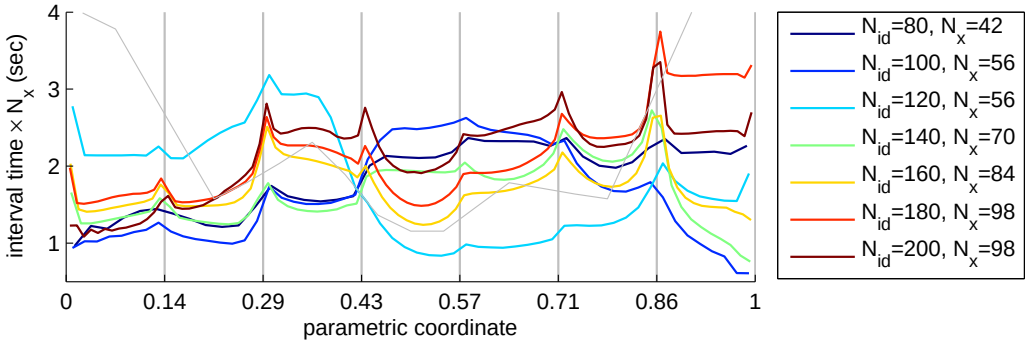


(a) $N_{id} = 20$ (b) $N_{id} = 40$ (c) $N_{id} = 60$ (d) $N_{id} = 80$ (e) $N_{id} = 100$ (f) $N_{id} = 120$ (g) $N_{id} = 140$ (h) $N_{id} = 160$ (i) $N_{id} = 180$ (j) $N_{id} = 200$

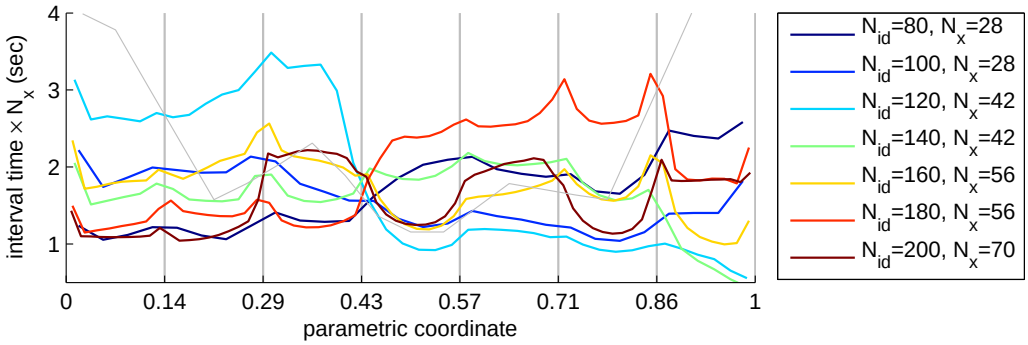
Figure 4.29: *Same as previous page, plotted at constant N_{id} values.*



(a) $N_{id} \simeq N_x$



(b) $N_{id} \simeq 2N_x$



(c) $N_{id} \simeq 3N_x$

Figure 4.30: *Several optimal timings around the magenta dashed lines of the plots* *p.115.*

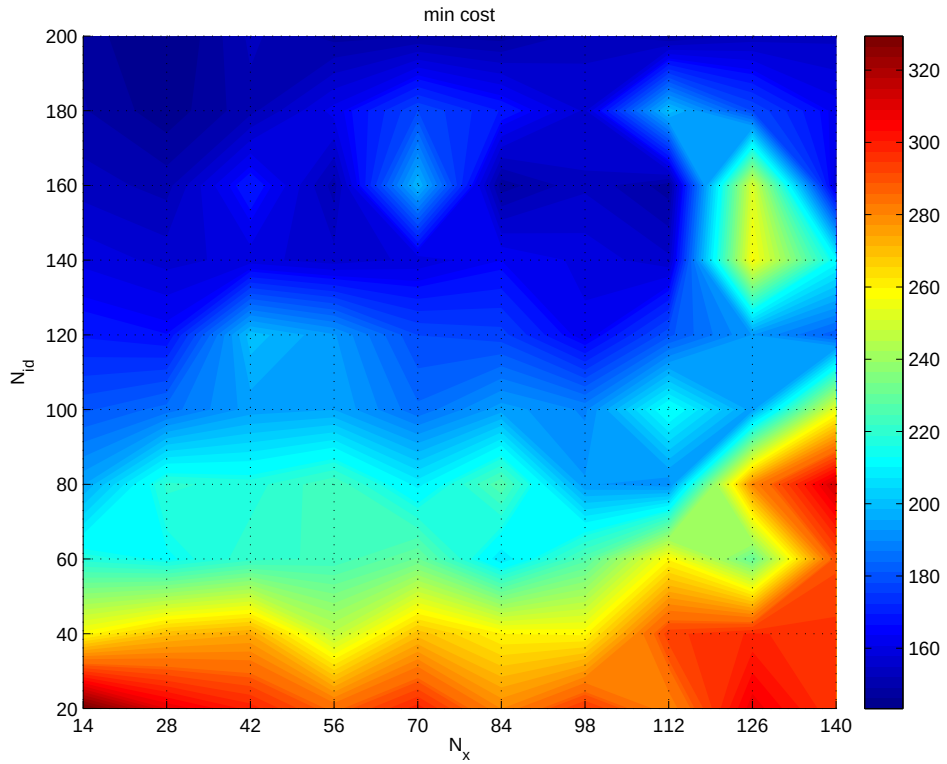


Figure 4.31: *Sampling study, minimal cost.*

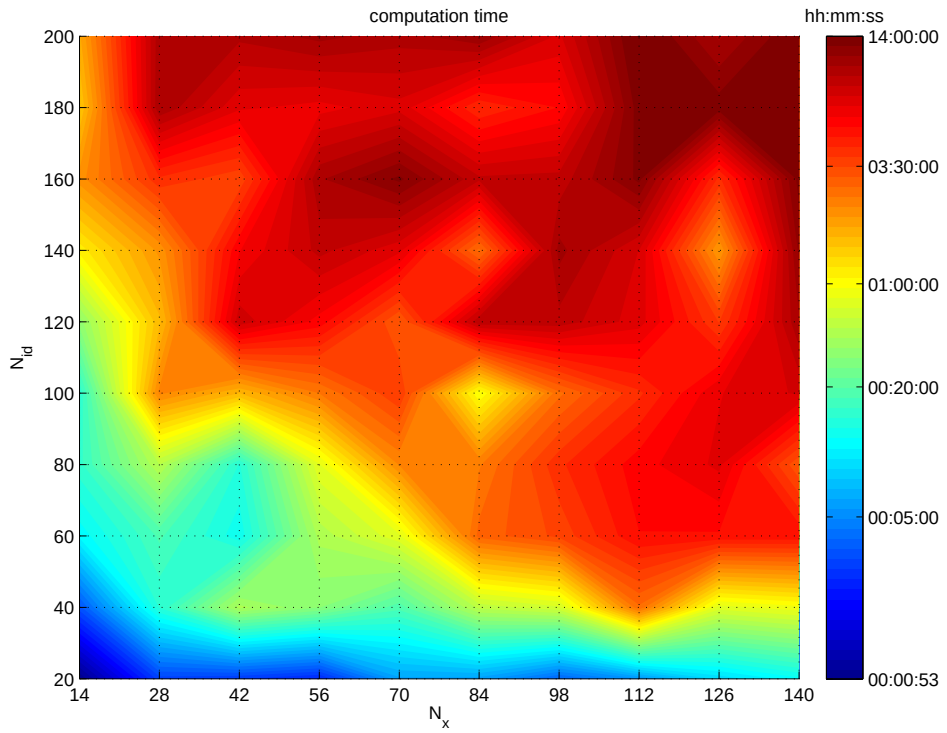


Figure 4.32: *Sampling study, computation time.*

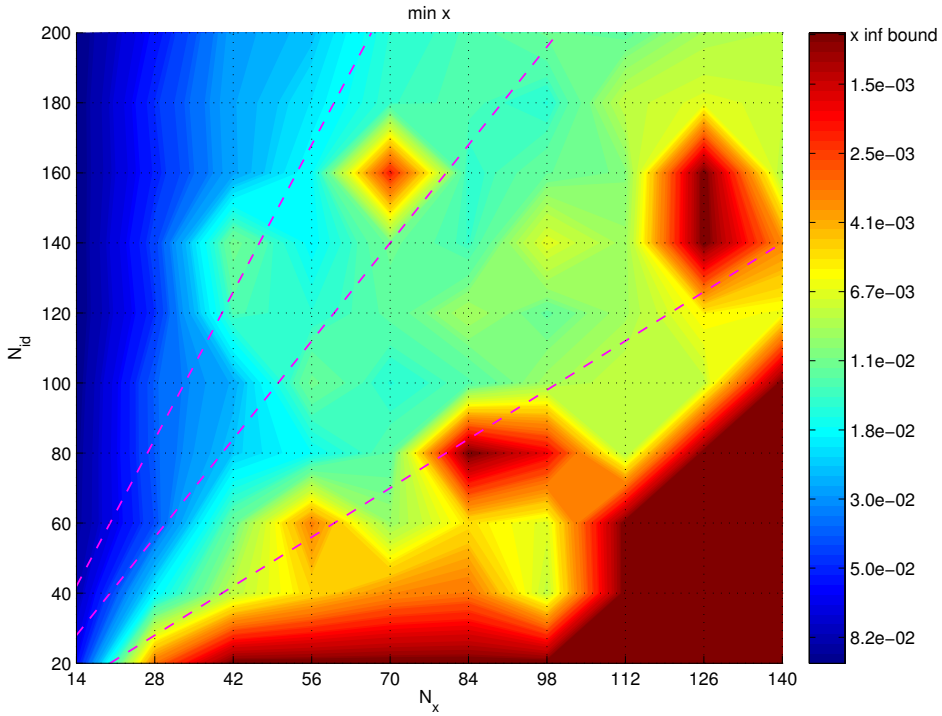


Figure 4.33: *Sampling study, minimal x-component value.*

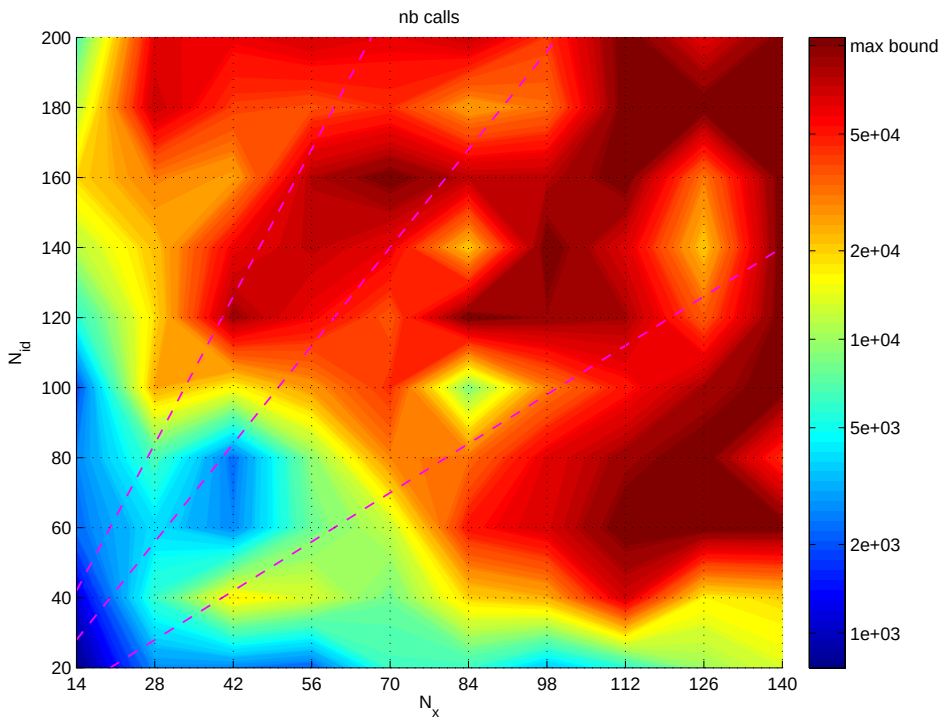


Figure 4.34: *Sampling study, number of cost function evaluations. Dashed lines correspond to the graphs p.113.*

figure 4.30a). Indeed with these values, some x components can be reduced up to the lower bound if the next components are increased, while preserving a smooth trajectory for the inverse dynamics frames. Some x components are therefore unconstrained and can move freely while the optimizer is running. We can conclude that by default, x has a tendency to diverge if it is uncontrolled. A slight regularization on x in the cost function could solve this issue.

The figures 4.28 to 4.30 surprisingly contradict the intuition that the thinner the samplings are, the better the solutions are, with an exception for the figure 4.29j, which gives the highest optimal timing similarities with $N_{id} = 200$ and $N_x \geq 56$. These results could be due to an error in the deployment of the program on the cluster, which has been done by flashing[1] the program. Another explanation could be that the thinnest samplings reach the machine precision, leading to aberrant optimal timings. A last hypothesis could be that the strong dependency of the cost function on the N_{id} parameter (see figure 4.31) leads to very different behaviors of the algorithm with respect to N_{id} variations. The verification of these three hypotheses is let for future works.

If one would have to choose two values, then $N_{id} = 200$ and $N_x \geq 56$ could be retained by observing the figure 4.29j, but the corresponding computation times are about 10 hours per optimization. Assuming these values gives roughly the timing shapes we want, the first N_x that gives an approximation of these shapes in the figure 4.28 is $N_x = 28$ (figure 4.28b). In this subfigure, the choice $N_{id} = 100$ seems to be the best compromise between the computation time and the approximation of the wanted timing shapes, which gives the final relations $N_x \simeq 4N_{keys}$ and $N_{id} \simeq 4N_x$ as the best choices.

Parameter setting conclusion

Quantitatively, the algorithm 1 gives the absolute best results, and also presents the technical advantage of being licence-free, which is better for the reproducibility of the results, and for cluster computations. About setting the parameters of this algorithm, the best choices are a spacing for the gradient computation $\Delta_g = 10^{-5}s$ and a stop criterion $\Delta_k = 10^{-5}s$. These values are inside the confidence region for the algorithm convergence, and gives the best results among the 81 candidates we evaluated. Finally, the constant interval time of $0.1s$ as starting point gives the best results among the four tested initializations, better than the approximated ground truth timing. The optimal timings have proved to be almost global optima. For the sampling, the best compromise is obtain for $N_x \simeq 4N_{keys}$ and $N_{id} \simeq 4N_x$. Further studies are needed to increase the robustness of the cost function to the N_{id} variations.

To conclude qualitatively this study, the setting of the input parameters could also have been the purpose of a big enclosing optimization, which would then have been the third level of optimization, taking into accounting the per frame least square solving for the cost function evaluations. Such an optimization scheme has

[1] flashing: copy past from one computer to another

already been used in [Liu 2005], but with only two optimization levels, where the aim was to determine the optimization parameters for which an input motion is the optimal solution. But each new stacked level of optimization brings also a new set of parameters that must be set, and at the end the setting of the top level parameters must be done by an external user. In the case of [Liu 2005], the advantage is that the top level has only a couple of parameters, whereas the lower one contains 147 parameters. Two solutions are conceivable to avoid an infinite stacking of optimization levels : the graphical study previously presented, and a self-optimizer, which would optimize its own parameters. Nevertheless, this last solution seems quite fancy as the initialization and the stop criteria can not be avoided.

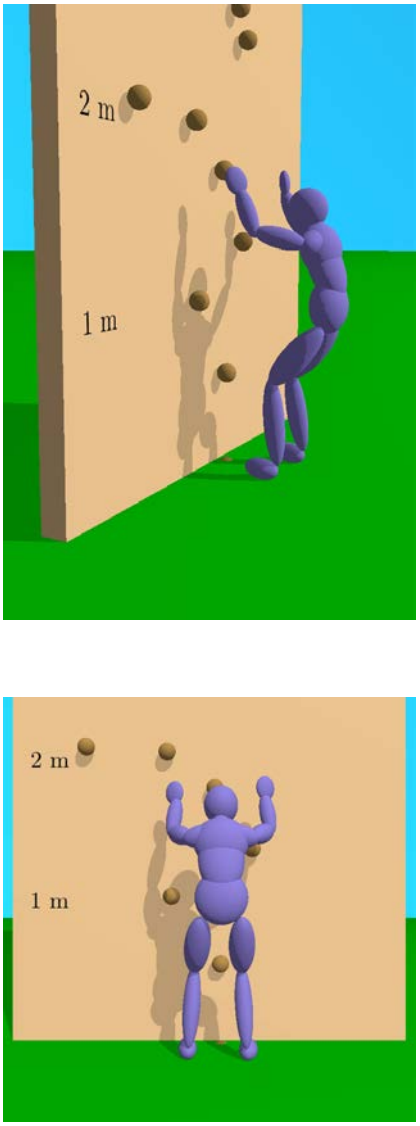
4.4.3 Long sequence

Once the timing algorithm and its associated parameters has been validated on a short sequence with available MoCap for keyframe generation, we turn to a longer sequence of a vertical climb of $10.5m$, without MoCap data. A focus is made on the validations of the aspects of this problem that differ from the bouldering example, which are the trajectory generation, a new effect of the trajectory sampling, and the segmentation of the optimization issue. Finally, once these new elements have been validated, a detailed ground truth timing evaluation is provided. This section is organized as follow:

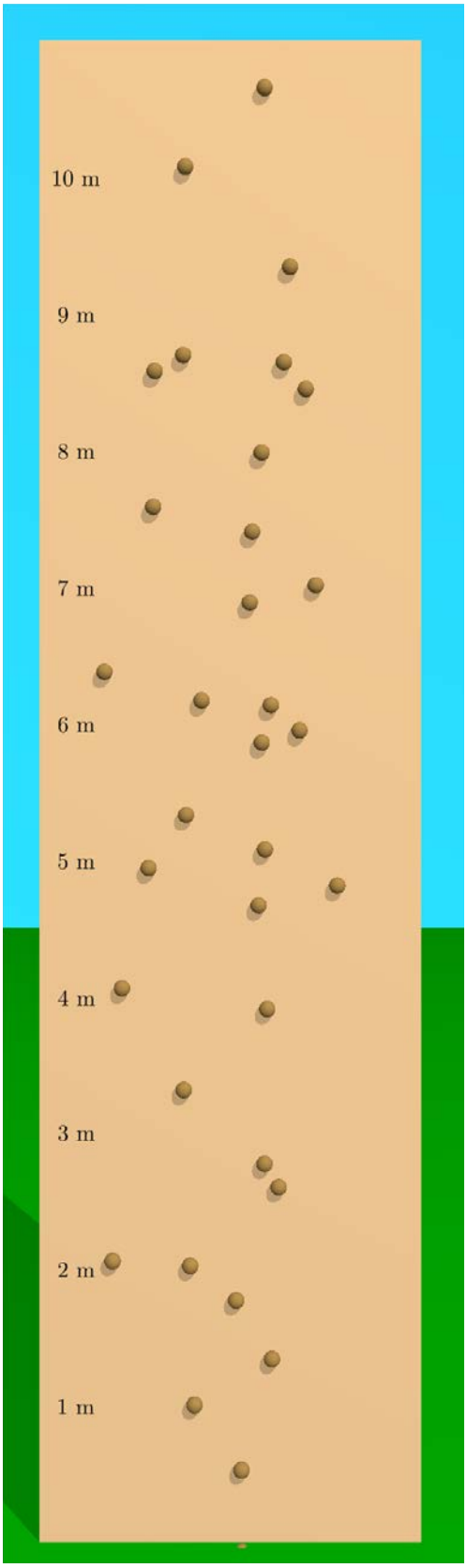
1. Input data presentation
2. Output overview
3. Generation of keyposes by inverse kinematics
4. Trajectory generation by keypose interpolation
5. Visual effect of the N_{id}–N_x sampling
6. Effect of cutting the optimization into segments
7. Ground truth timing comparison

Input data presentation

The input data for the long sequence experiment are on one hand, the climber, and on the other hand, the route to climb. The climber is the same as the one used in the previous bouldering experiments. He has the same local contact points and has the default pose plotted in the figure 4.35a, representing the default climbing posture. The route contains a set of holds, as shown in the figure 4.35b, and the order in which these holds should be taken, as depicted in the table 4.6. These quadruplets of holds to take has been numbered from A1 to G3, A1-A7 being the first 7 sets of holds, B1-B7 being the sets 8 to 14, etc. This sequence includes two hand matchings



(a) *The climber.*



(b) *The climbing route.*

Figure 4.35: *Inputs: (a) climber's default pose, and (b) external contacts.*

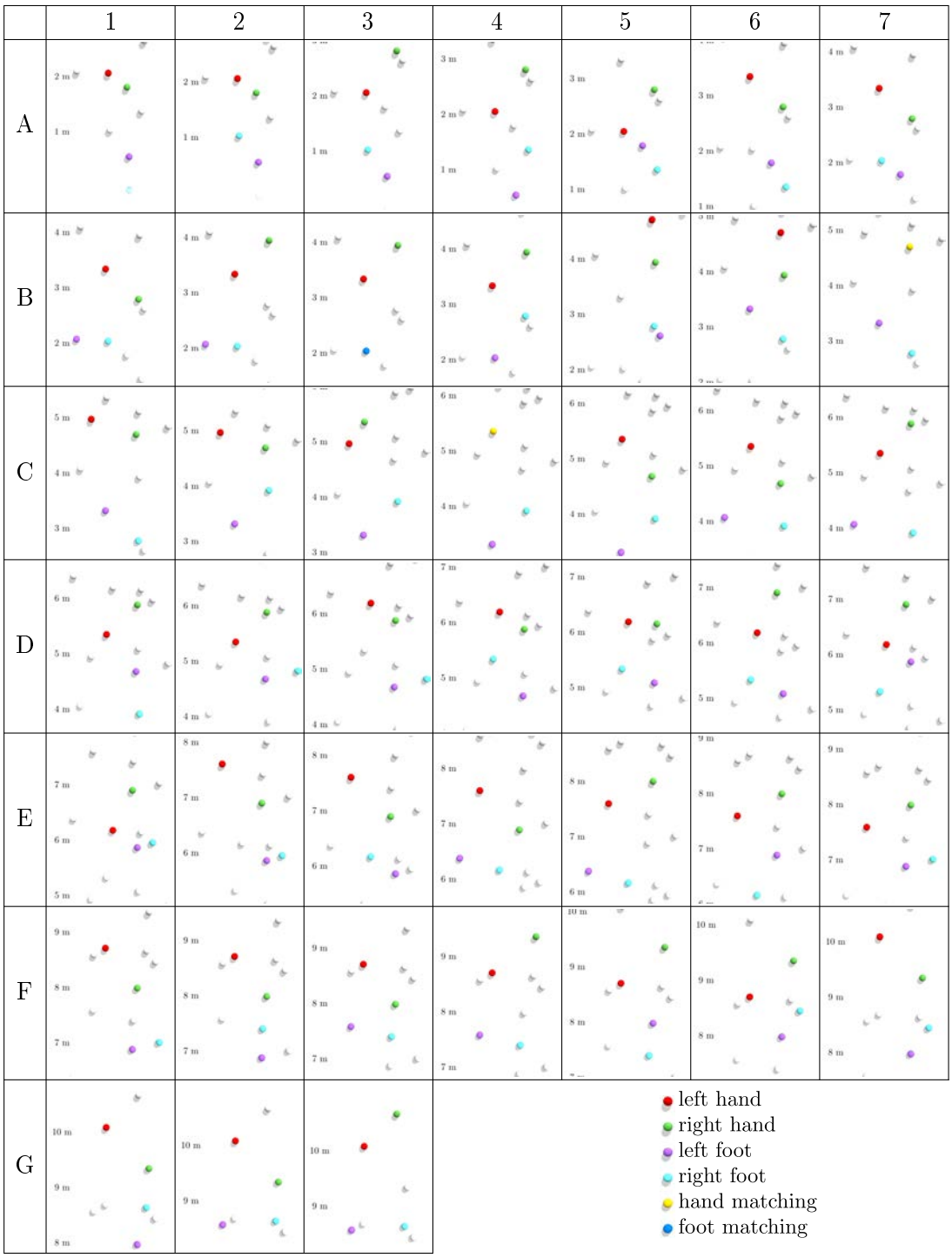


Table 4.6: *The input contact sequence for the Espace Vertical example (row-wise ordered).*

(cells B7 and C4), one foot matching (cell B3), two double contact moves (B4-B5 and D4-D5), and several foot cross-overs (e.g. A2-A3, D4-D7), reflecting the multiplicity of the movements encountered in rock climbing.

Output overview

The output of our algorithm is displayed figure 4.36. As for previous the results, each frame has been translated toward the right, producing an artificial horizontal displacement that does not exist in the original route trajectory. At each frame, the camera is vertically traveled to follow the climber, also producing an artificial effect of a treadmill motion. In this figure, we can see a fluent ascent of the route (no discontinuity in the motion), with a fast but plausible total time of $13.5s$ for the $10.5m$ of the route. Indeed, this time is 2.5 times slower than the 2012 world record in speed climbing, which is $5.88s$ for $10m$, showing the realism of this computed optimal timing.

In the following, we provide a thinner evaluation of this result, by tackling in the following order the items

- Generation of keyposes by inverse kinematics
- Trajectory generation by keypose interpolation
- Visual effect of the N_{id}–N_x sampling
- Effect of cutting the optimization into segments
- Ground truth timing comparison

Generation of keyposes by inverse kinematics

The climber's keyposes are obtained by inverse kinematics. This process starts by positioning the CoM of the default pose at the center of the 4 holds to take. Then this CoM is translated 25 cm away from the wall. Finally, the four limbs of the climber are adjusted such that the hands and the feet touch the holds to take. The table 4.7 shows the results of this process. These results can be evaluated with the ground truth shown in the table 4.8. By comparison of these two tables, we notice that the ankles are well adjusted to recover the full extension poses (A3, A4, F7, G1). The backstepping leg positions are also well recovered (e.g. E2, E7, F6), as well as the leg orientations for the foot cross-over sections (A2-A3, D4-D7). On the other hand, the head movements during the observation phases are missing (e.g. B3, B6, F2), and the torso position and orientation is not always accurately recovered (e.g. C6, E1, F3), leading to incorrect leg and arm configurations.

Trajectory generation by keypose interpolation

The keypose interpolation is done in the three steps presented figure 4.37. First, keyposes are interpolated with quaternion cubic splines, which choose by default the

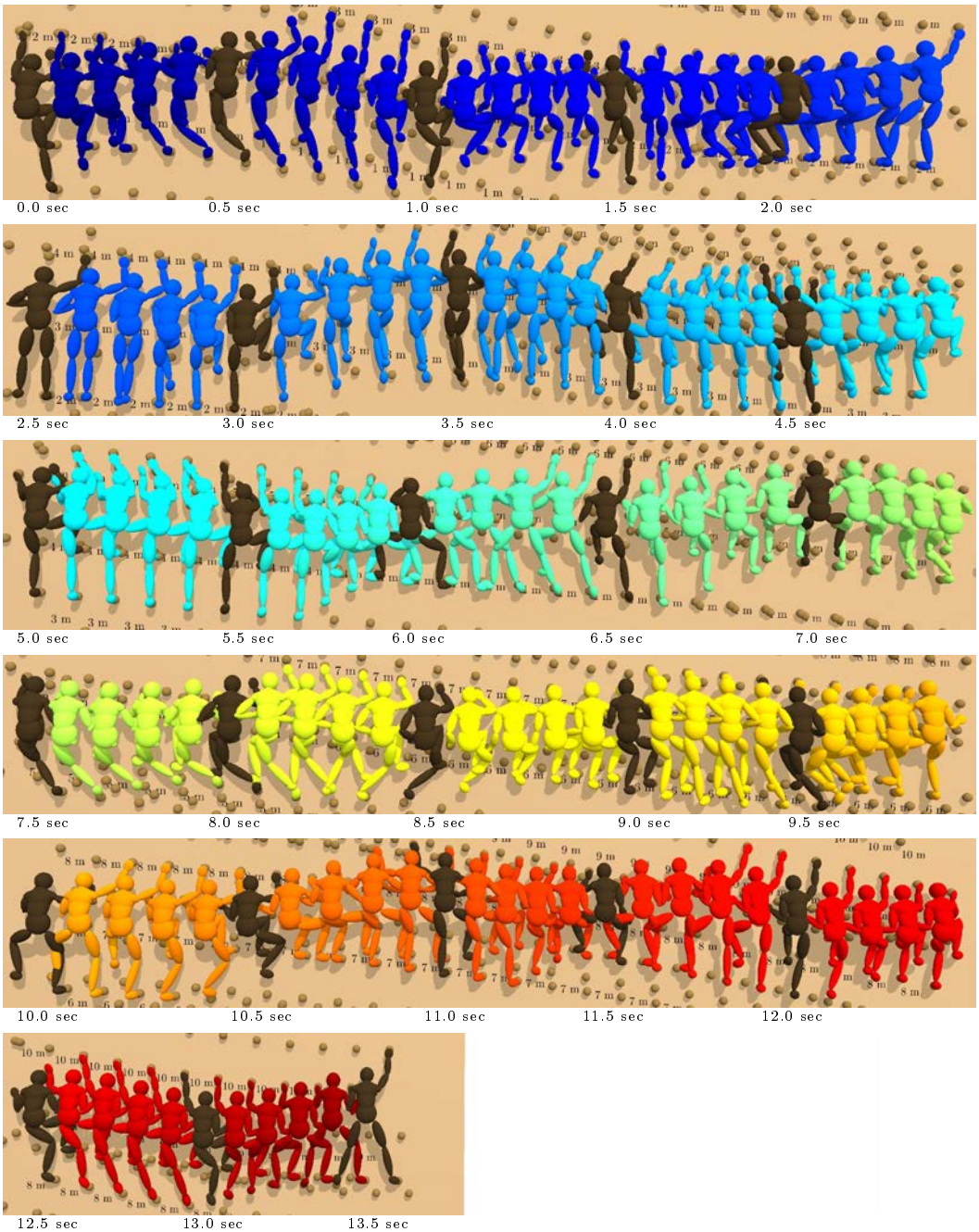


Figure 4.36: *Physically optimal timing for the Espace Vertical sequence. The horizontal axis is the time. The brown poses are time spacing marks of 0.5 second. The camera is traveling vertically to follow the climber.*

Table 4.7: *The output keyposes, computed by inverse kinematics on the contact sequence.*

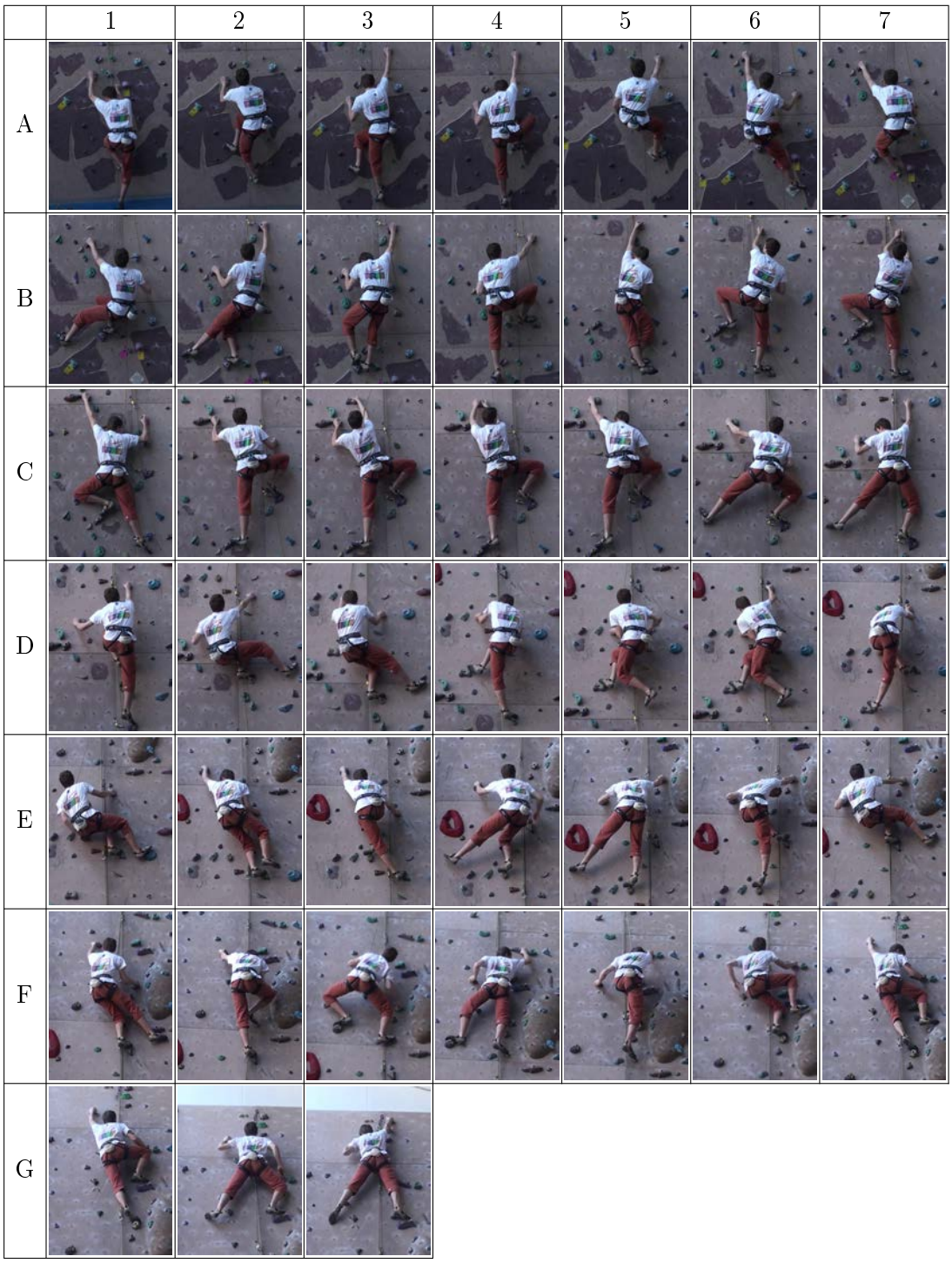
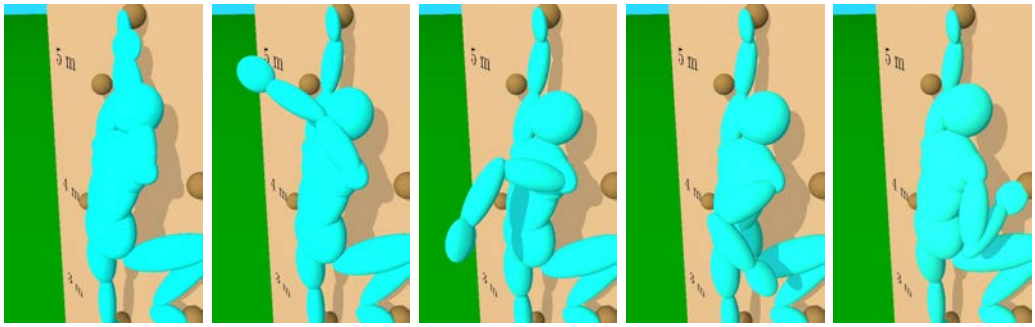
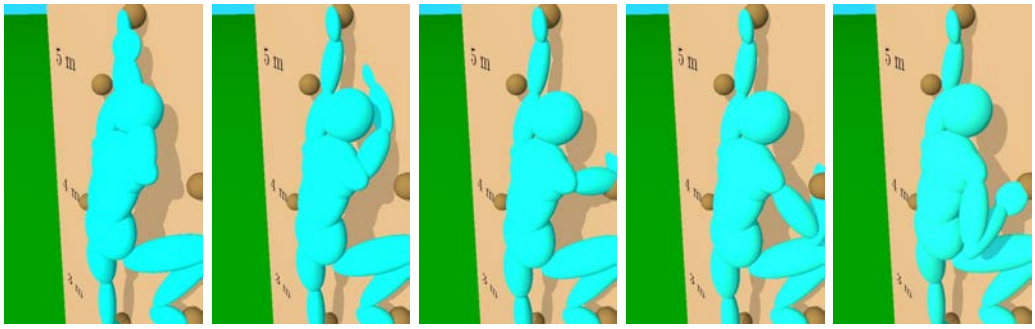


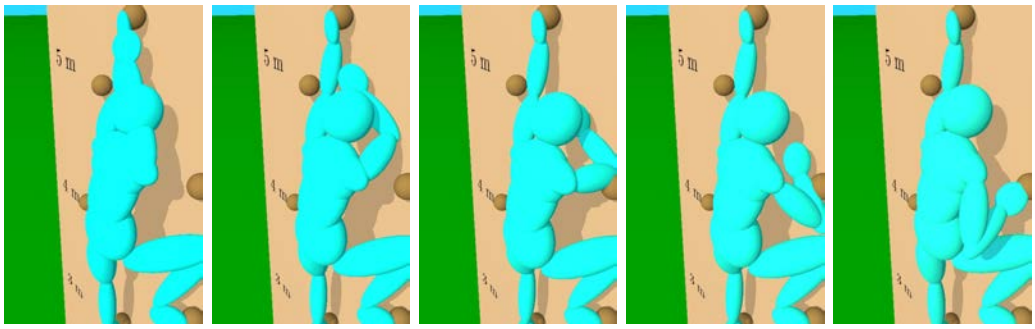
Table 4.8: *Ground truth for the inverse kinematics output.*



(a) *Shortest path for quaternion cubic spline interpolation of keyposes C4 and C5.*



(b) *Longest path interpolation for the right shoulder.*



(c) *Contact linear interpolation.*

Figure 4.37: *Steps of the keypose interpolation with the move C4-C5.*

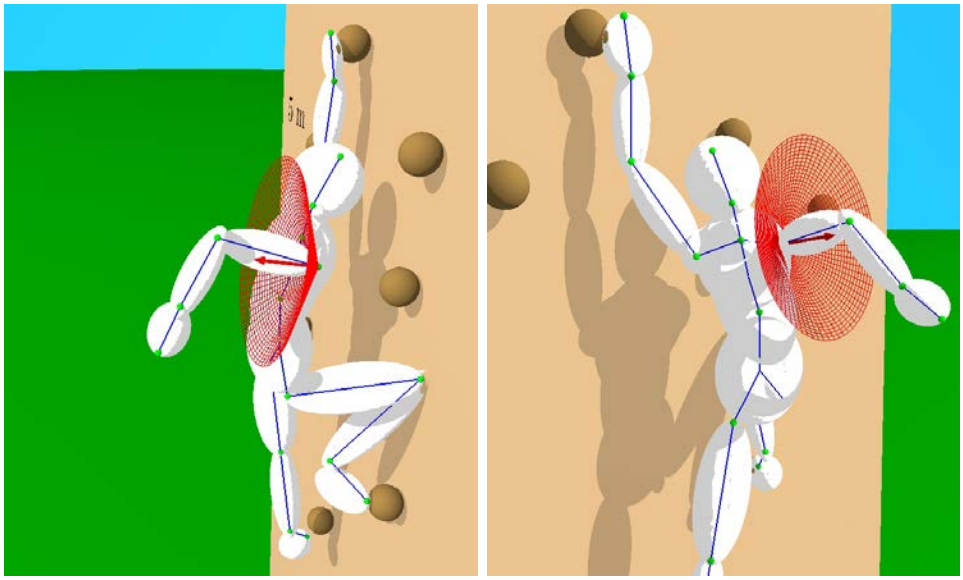


Figure 4.38: *Joint angular limit cone. The red arrow is the axis of symmetry of the cone. The angle between this axis and the upper arm axis must be higher than the cone angle.*

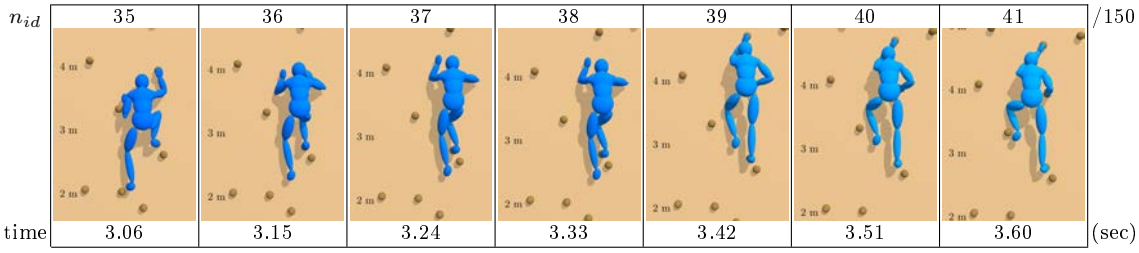
shortest path between joint orientations. Then angular joint limits are controlled on shoulders and on hips with a limit cone, as in figure 4.38, and the interpolation is recomputed with the longest paths where joint dislocations have been observed. Finally, we linearly interpolate the world position of the local contact points, guided by the previous interpolation, to avoid the hands and the feet to break through the wall.

The difference between the angular limit cone we used and the axis/twist angular limits of [Macchietto 2009] is that we do not use the identity transformation as the center of the allowed angle cone. Indeed this would make the coronal plan[2] of the climber a symmetric plan for the shoulder range of motion, but in practice it is not the case. The right elbow can almost touches the left shoulder when passing ahead of the torso, but is far from touching it when passing behind the torso. Moreover, despite the axis/twist angular limits allow thinner representation of joint angular limits, the limits we used based on the angle between the upper limb axis and the cone axis are more intuitive to set up, which is important if the end user is an artist.

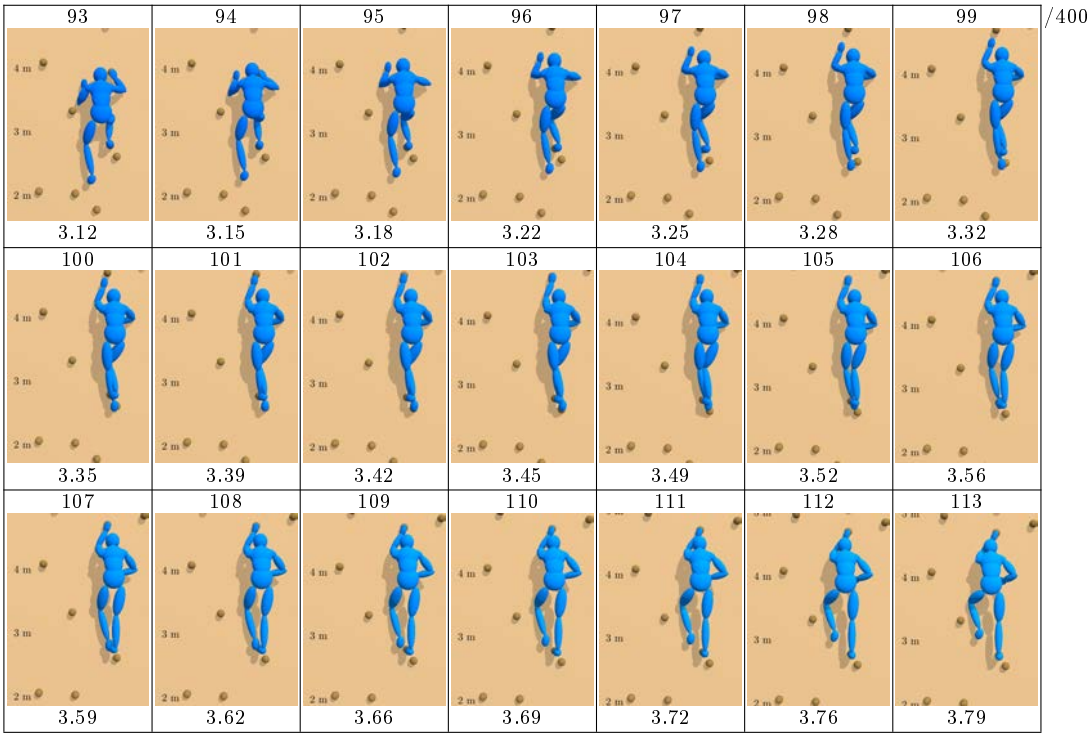
Visual effect of the N_{id}–N_x sampling

This subsection shows that $N_{id} = 150$ leads to an unwanted visual artefact, avoided with $N_{id} = 400$. The figure 4.39 presents the frames used for the cost function evaluation with these two total numbers of cost frames at the optimal timing obtained with $N_x = 132$. In the figure 4.39a, the frames 37 and 38 are identical,

[2]coronal plan: vertical plan containing both shoulders



(a) *Using* $N_{id} = 150$.



(b) *Using* $N_{id} = 400$.

Figure 4.39: *Visual comparison of two inverse dynamics samplings for moves B4-B6.*

showing that it is less costly to stop and then to do a big step between frames 38 and 39 than to pass by the extremal position B5, where the left foot touches the wall. This phenomenon constitutes a second candidate to explain the timing noise previously observed (figure 4.30a). If the final time step is 25 fps for instance, the in-air slow down at N_{id} indices 37 and 38 is noticeable, as between frame indices 36 and 40, which is roughly the range of influence of the slow down, 0.36 second have passed, leading to 9 output video frames. The higher sampling rate figure 4.39b where $N_{id} = 400$ resolves this artefact. Here the sampling does not allow anymore to avoid the keypose B5. The climber must then slow down on the whole trajectory segment between keyposes B4 and B6 to be able to take an impulse on the left foot at keypose B5 in order to change its trajectory for the next move. This higher sampling rate thus lead to a more natural output timing.

Effect of cutting the optimization into segments

Optimizing the timing on the whole climbing route with the parameters of figure 4.39 for instance took respectively 5 h 15 for figure 4.39a and 6 h 40 for figure 4.39b. To reduce this computation time, the optimization is segmented, as in [Al Borno 2012] for instance. The originality of our study is to evaluate the quality of the segmented optimization with respect to the corresponding all-at-once optimization. In the figure 4.40, the timings for the moves B4-B5, D1-D2 and E5-E6 are computed with an increasing segment width from 1 keypose interval to 13 keypose intervals. All results have been computed with $x_0 = 0.1s$, except for the move E5-E6 with 9 keypose intervals, which is the only case where several initializations have been tested to finally choose $x_0 = 0.09s$, as showed figure 4.43. By zooming on the moves of interest (figure 4.41) we see that timing for the central moves are similar for 7 keypose intervals and more. Moreover, as shown in the figure 4.42, the computation time drastically increases between the interval sizes 7 and 9, from $20min$ to $1h$. Therefore, the segmentation size of 7 keypose intervals is chosen for the segmented optimization, whereas in [Al Borno 2012], two successive windows of $0.5s$ are arbitrarily chosen to constitute each optimization segment.

Optimal timings computed in a single large optimization are plotted figure 4.44. Their related computation times are gathered in the table 4.9. The optima are quite close to their initializations, except for the keypose B5. On the contrary, segmented optimizations lead to much better convergences, as plotted figure 4.45, with the corresponding computation times gathered in the table 4.10. These last optima still contain some outliers, highlighted in yellow, that can be removed by a post-selection as used for the figure 4.46. This post-selection is made possible due to the overlapping segments. Indeed for each move, the timing for the 3 previous moves and the 3 next moves are also computed. The timing of every move is thus computed 7 times, with more or less accuracy, as shown in the cyan curves of figure 4.40 : the farthest we are from the central move, the less accurate is the timing. We thus use the 3 central moves to compute outlier timings, by choosing among those candidates the timing the closest to the mean timing of those candidates.

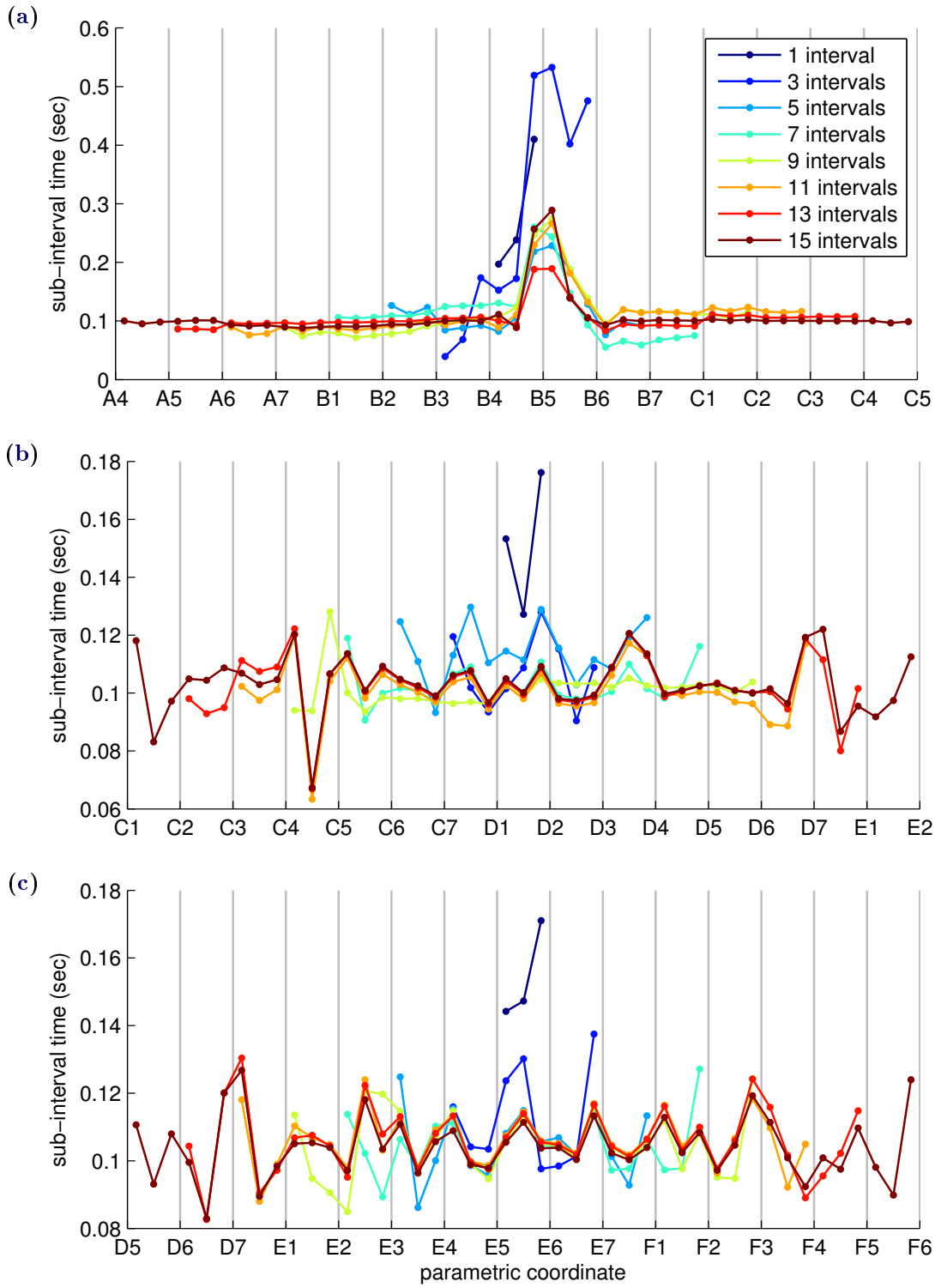


Figure 4.40: *Optimal timings for moves B4-B5, D1-D2 and E5-E6 with several segment widths.*

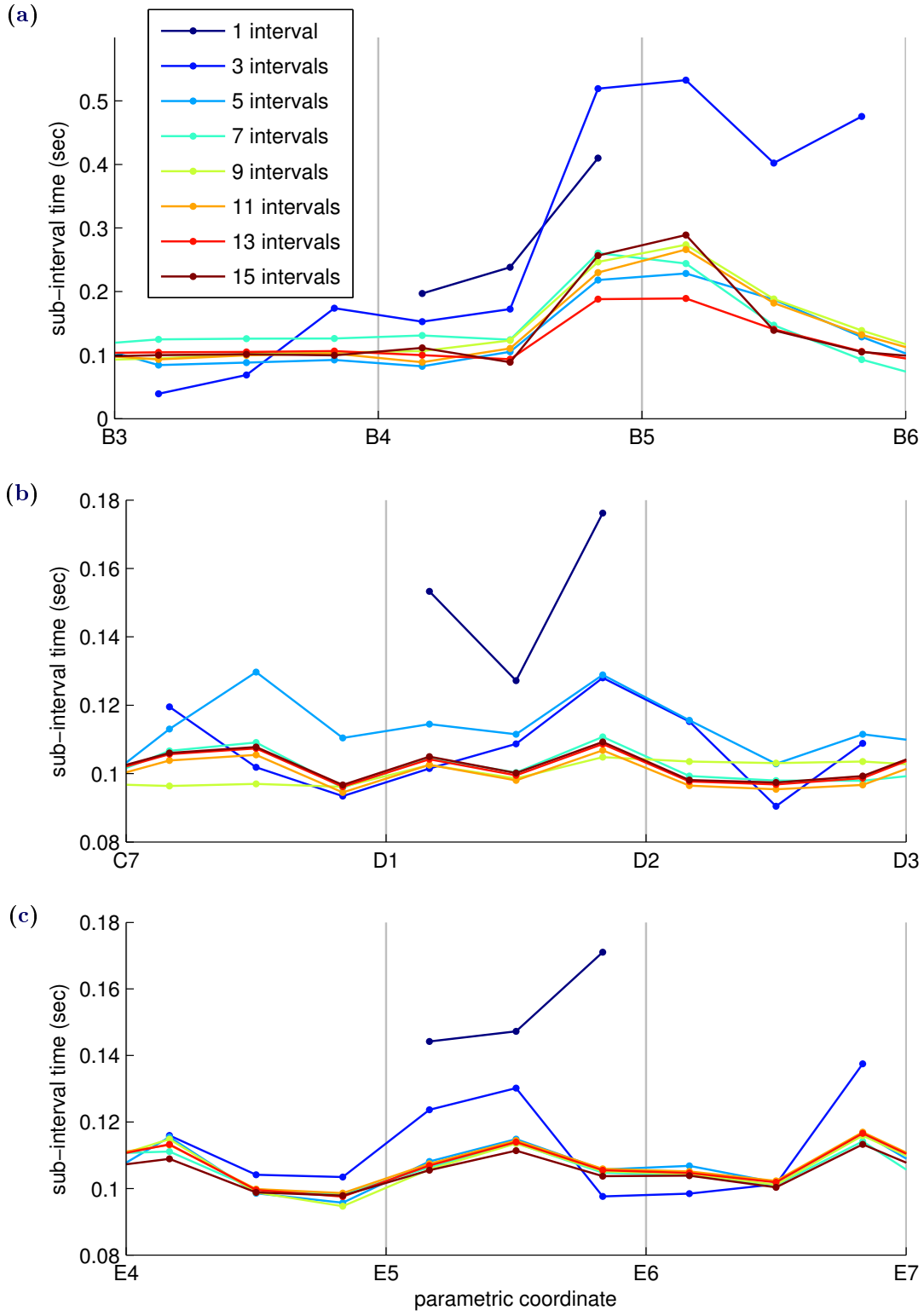


Figure 4.41: *Close up of the previous results on the segments of interest.*

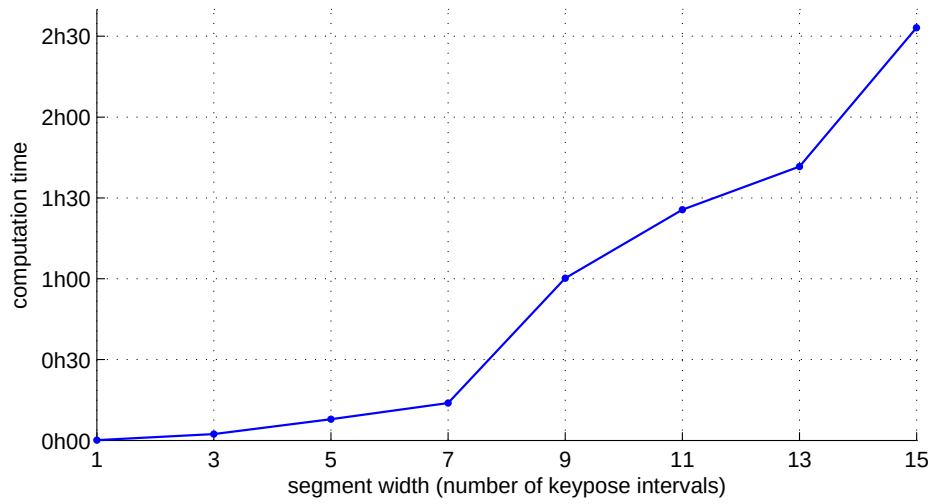


Figure 4.42: *Average computation time for each segment width.*

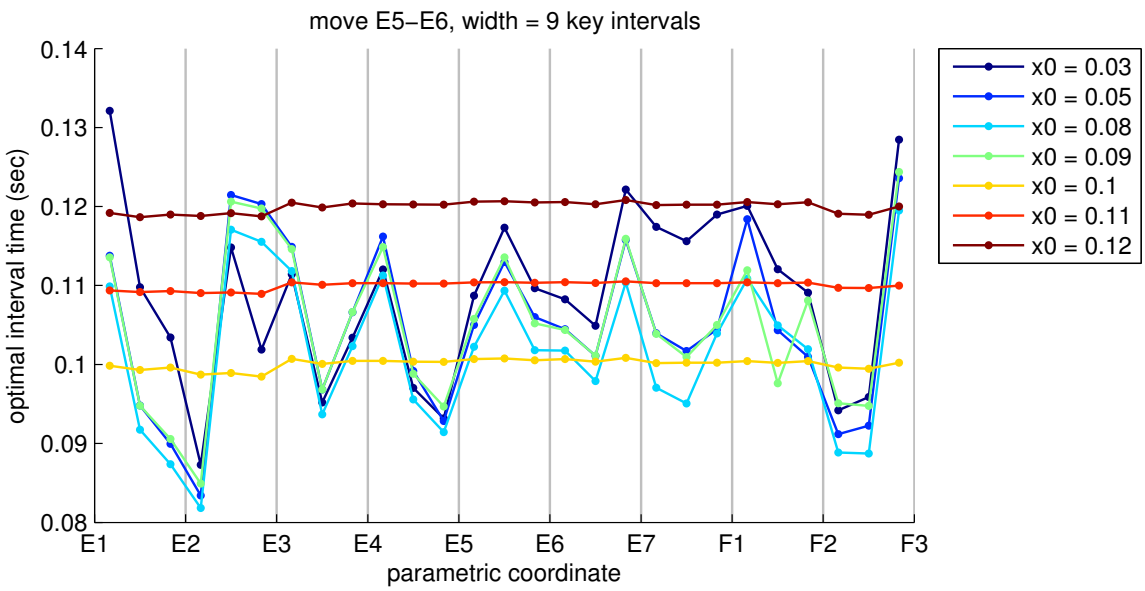


Figure 4.43: *Several initializations for move E5-E6, width 9 intervals.*

x_0	0.03	0.13	0.15	0.1
Computation time	3 h 57	11 h 24	1 h 01	6 h 42

Table 4.9: *Computation time for the timing optimization of the 44 moves, computed all-at-once on a 3.3 GHz computer.*

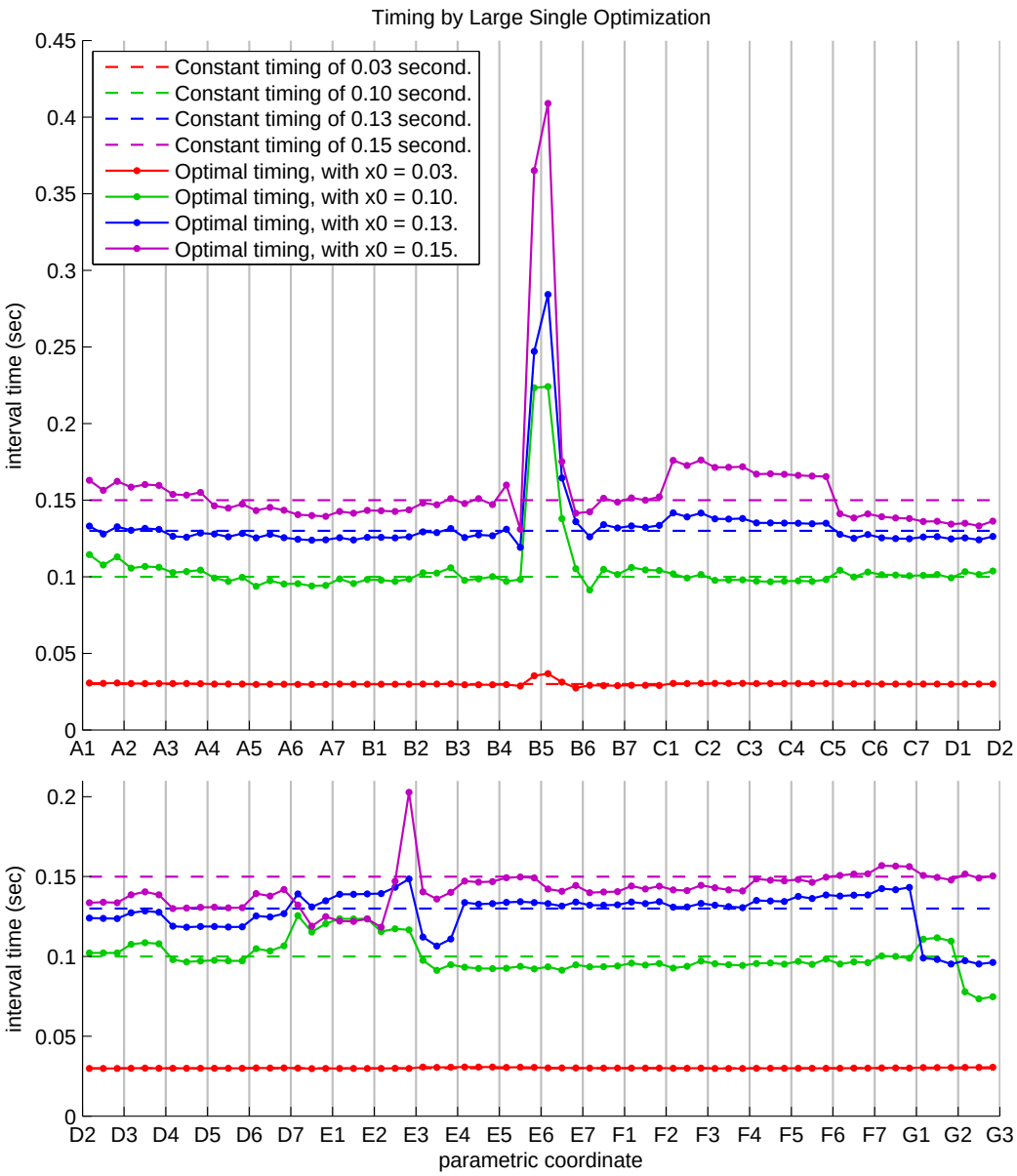


Figure 4.44: *Espace Vertical optimal timings computed all-at-once.*

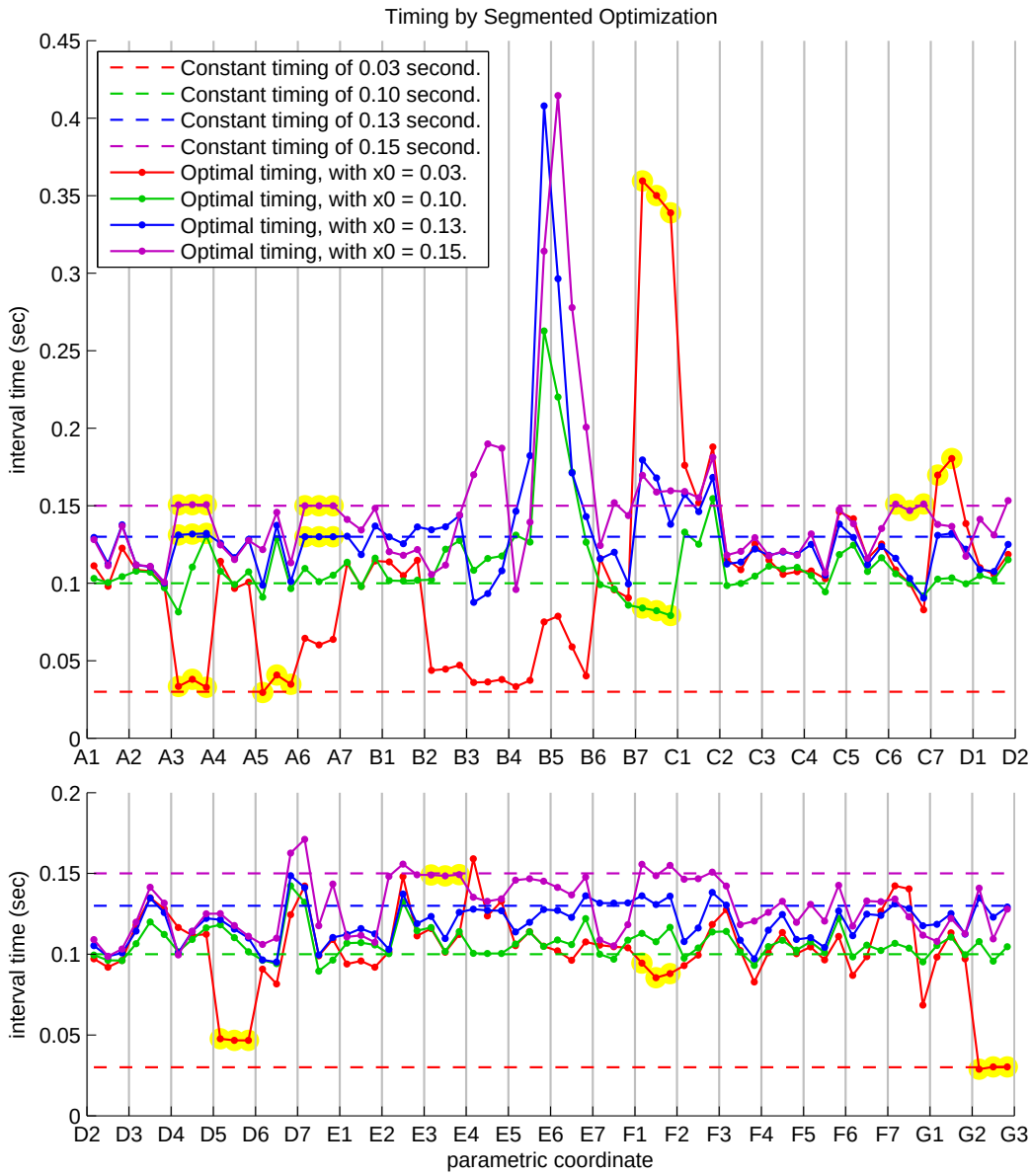


Figure 4.45: *Optimal timings computed by segments of 7 intervals.*

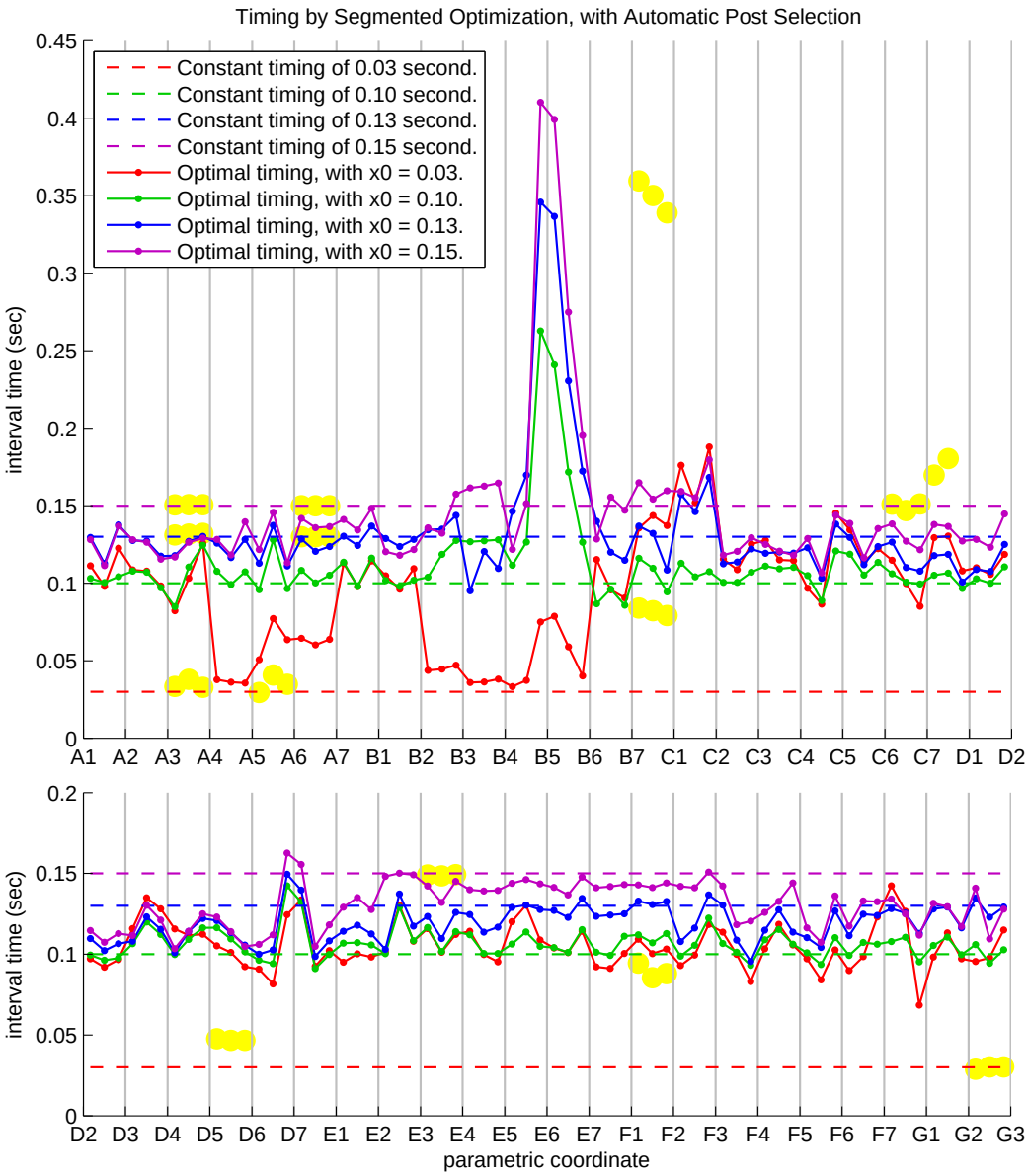


Figure 4.46: *Segmented timing optimization with post interval selection.*

x_0	0.03	0.13	0.15	0.10	0.10
Average time	1 h 06	35 min	35 min	22 min	22 min

Standard deviation	45 min	24 min	26 min	16 min	15 min
Minimum	1 min 23	1 min 01	53 sec	1 min 33	1 min 35
Maximum	3 h 02	1 h 34	1 h 57	1 h 15	1 h 06
CPU speed (GHz)	2.4 (66%), 3.3 (20%), 2.7 (14%)			2.4	3.3

Table 4.10: *Computation time statistics for the timing optimization of the 44 moves, computed individually.*

Ground truth timing comparison

The average of the four optimal timings found previously by segmented optimization is compared to the ground truth timing in the figure 4.47. In this figure, a clear slowing down can be seen for the moves B4-B6. Two hypotheses can be established to explain this slowing down. First, we may argue that the move B5-B6 is more efficient when using a ballistic motion, i.e. when starting quite fast and then letting the gravity change the body velocity without making much effort against it. Indeed, this is the only large move with only two contact points, which are moreover on the same side of the body (right hand and right foot). For this particularly difficult move, a ballistic move is thus more needed than for the others, less difficult. The second explanation is found at the pose B5 itself, where both the torso and the left leg must change their velocity direction by 90°. Indeed, the main trajectory of the move B4-B5 is a 45° right diagonal ascent, whereas the main trajectory of the move B5-B6 is a 45° left diagonal ascent. This drastic change of direction requires a slowing down to avoid a high acceleration at the keypose B5, which would be very costly in term of joint torques.

Still in the figure 4.47, the optimal timing is faster than the ground truth timing. This difference can be mainly explained by contact establishment latency and hesitation or observation phases, as shown figures 4.48 to 4.52. In these figures, the time spent by the climber in those phases is delimited by the red time instants. This time is carried over to the table 4.11, where a comparison is made with the timing differences between the optimal timing and the ground truth timing. These last temporal differences are interval time between sub-keyposes. As there are three sub-keypose intervals per keypose interval, we multiplied them by three to obtain keypose interval time as unit. In the table 4.11, the three contact establishment latencies observed in the video correspond to the differences between the optimal timing and the ground truth timing. For the two hesitation phases, the video time differences correspond to two thirds of the graph time differences. We hypothesize that the last third of these differences could correspond to the contact establishment at the two extremities of the corresponding keypose intervals. These five timing differences show that up to the contact establishments and the hesitation times, the

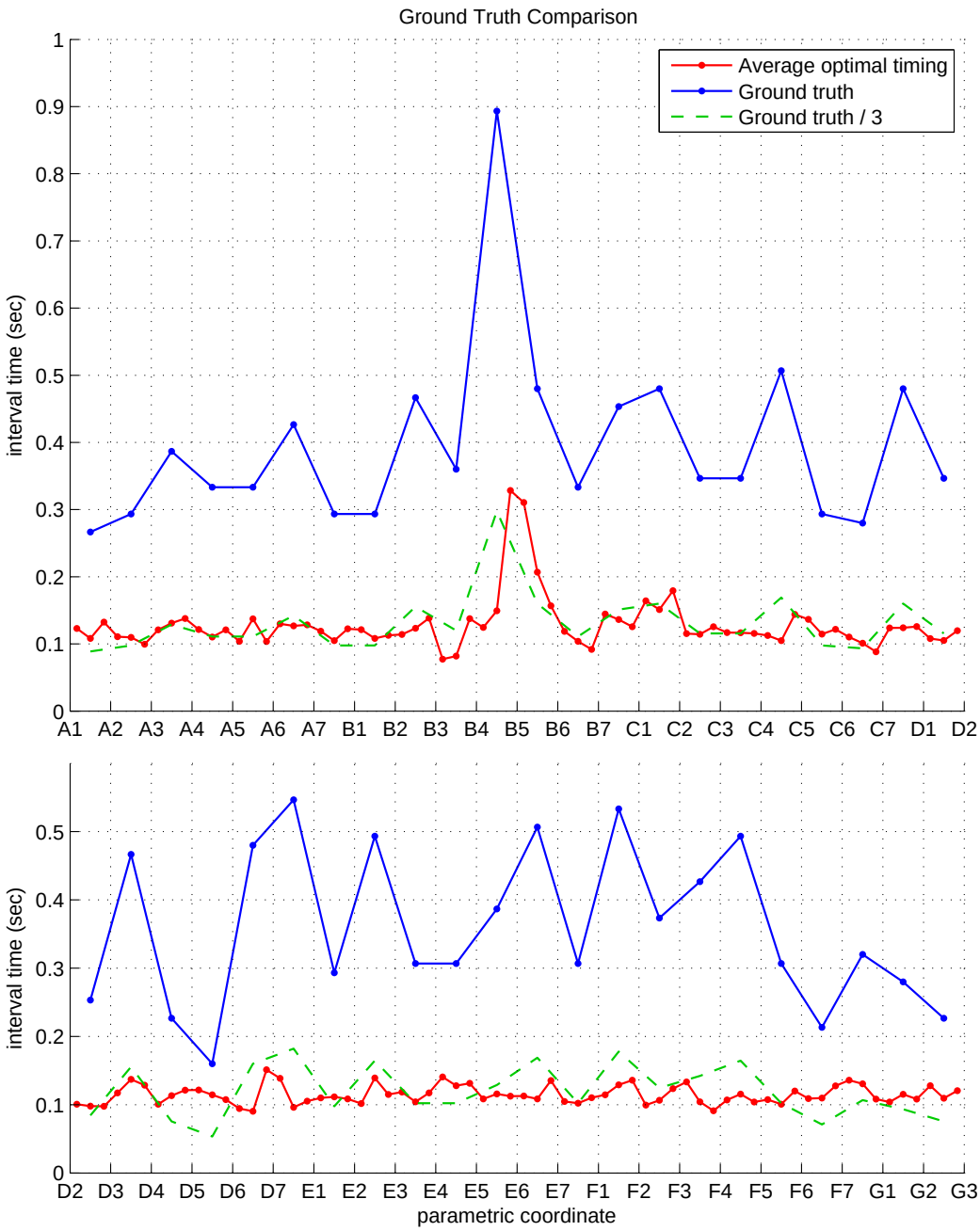


Figure 4.47: *Optimal timing and ground truth comparison.*

Figure 4.48: *Contact establishment at pose B2. The numbers are time instants in second in the original video. The red numbers are the start instant of the establishment, when the hand touches the hold, and the end instant, when the foot moves.*

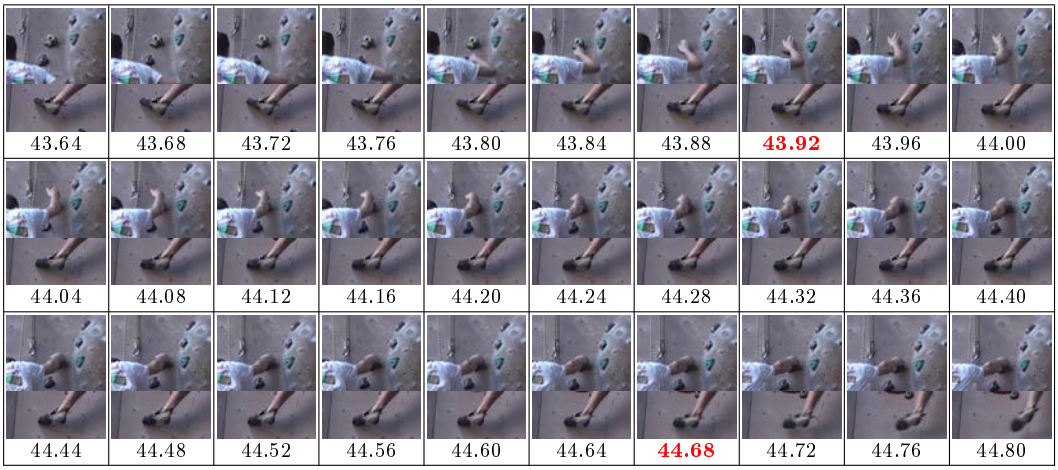


Figure 4.49: *Contact establishment at pose E5.*

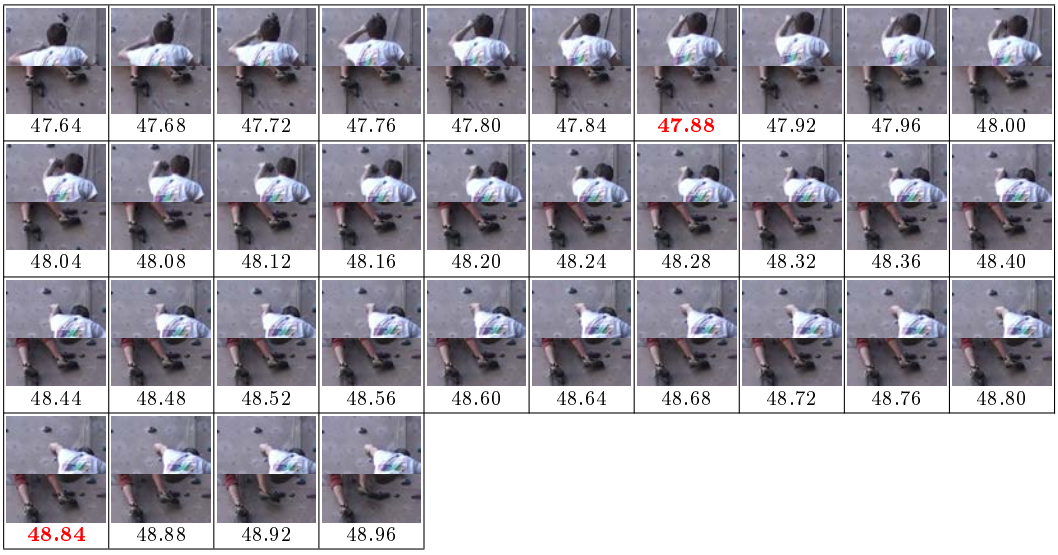


Figure 4.50: *Contact establishment at pose F1.*

Figure 4.51: *Hesitation phase for move C4-C5 : The right hand goes toward a hold, but does not take it.*

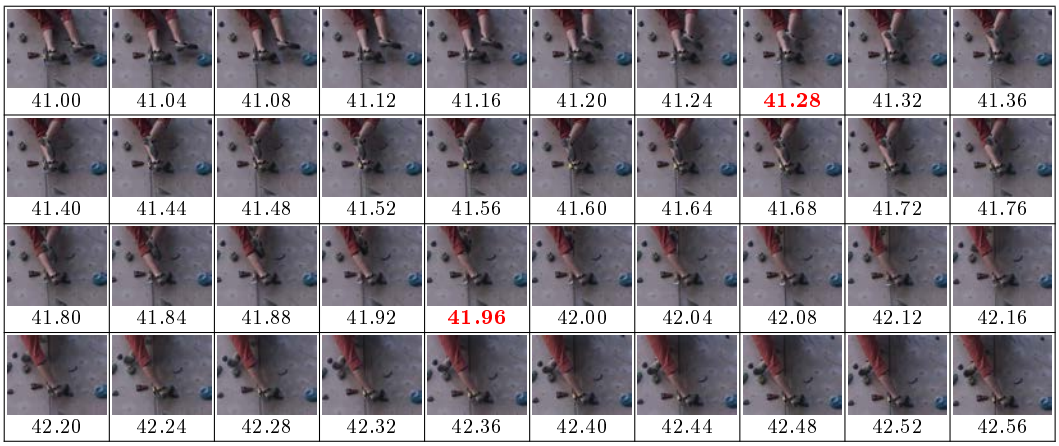


Figure 4.52: *Hesitation phase for move E2-E3 : The right foot stops behind the left leg, and then passes in front of it.*

(a) *D5-D6: smearing* **(b)** *F4: undercling* **(c)** *F6: inefficient move*

Figure 4.53: *Close-up on three particular keyposes to explain three differences between ground truth / 3 and our results.*

optimal timing is close to the observed timing.

To be able to validate the overall shape of the optimal timing, we assume that the contact establishment time, which corresponds to a fraction of the time when both hands and both feet are in contact, is proportional to the time spent in-between contacts, when a hand or a foot is moving. This has already be suggested by [Guidi 1999] who mentioned that the proportion of effective motion and other instants are about 30% and 70% of the total climbing time respectively, based on the work by [Dupuy 1989]. Therefore, we divide the ground truth timing by 3, in order to suppress the 70% of no-motion time that we do not modelize, and to focus on the time during which the climber is moving. In the figure 4.47, this comparison highlights another kind of differences at move D5-D6 and at keyposes F4 and F6. These instants are shown figure 4.53. The two first differences are due to the hold modeling. Indeed, the hold models are cost-free bilateral contacts for the hands and unilateral contacts with a friction coefficient of 0.8 for the foot, with the same

Instant	Figure	Observed latency (s)	Difference with the optimum (s)	Explanation
B2	4.48	0.96	0.9	Contact establishment
E5	4.49	0.76	0.75	
F1	4.50	0.96	0.9	
C4-C5	4.51	0.76	1.2	Hesitation
E2-E3	4.52	0.68	1.05	

Table 4.11: *The latency observed in the video and the corresponding delay between optimal timing and ground truth timing in the figure 4.47.*

contact normal for all the foot contacts corresponding to a vector bent at 30° from the vertical axis, towards the wall normal. This contact model is erroneous for the move D5-D6 (figure 4.53a) where the left foot of the climber is smearing the wall, i.e. it pushes the wall where there is no hold using only the friction between the wall and the shoe to progress, and for the keypose F4 (figure 4.53b), where the left hand takes an undercling hold, i.e. a hold oriented in a downward direction. For both these instants, the contact normal should be respectively normal to the wall and in a downward direction. These modeling errors are then propagated to optimal timing errors. The third difference (figure 4.53c) highlights an inefficient move corresponding to a faster ground truth / 3 timing than the optimal timing. Indeed, at this instant, the climber has released his left hand before having put his right foot on the wall. The move is thus done with two contact points instead of three. The strength needed by the two limbs involved in these contacts is thus higher than the one that would have been needed if a supplementary limb was available to help supporting the climber for this move. In this case, the optimal timing is thus better than the observed timing. Therefore, the differences between our optimal timing and the ground truth / 3 timing are explained by contact modelization inaccuracies and non-optimal ground truth.

4.5 Conclusion

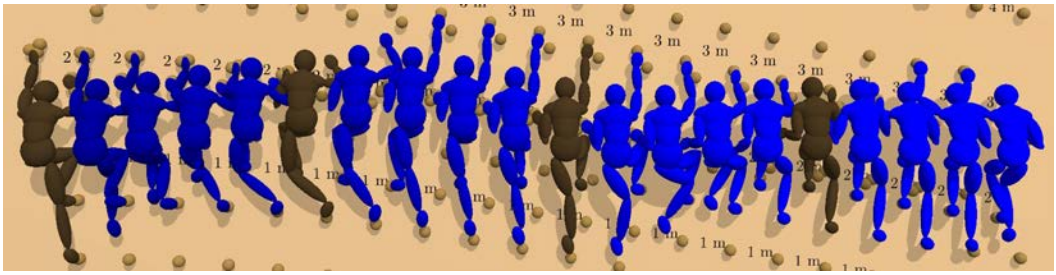
In the case of climbing motions, we showed that a realistic timing can be synthesized from the minimization of internal joint torques and the penalization of in-air contact forces, providing a continuous contact model. Moreover, the motion restrictions brought by the specific environment on which the climber evolves, allows to modelize realistic climbing paths by procedural inverse kinematics and quaternion cubic spline interpolations. A ground truth comparison based on a video sequence showed the accuracies and limitations of our approach, which is accurate for movements in-between contacts, but lacks some human factors, such as contact transition models or a decision making process.

The contact establishment and breaking latency observed in the previous chapter is confirmed in the case of motion synthesis, although the difference between contact transitions and observation instants is not obvious. On one hand, the contact change latency could be due to the need for the fingers to be precisely adjusted to the shape of the holds at the hand contact establishments. This precision necessitates some finger perception feedback involving a time consuming loop between the finger touch, the brain stimulation and the neuronal back response toward the finger muscles for readjustment. This loop could be simulated by random or constant delays of a few milliseconds of immobility at each contact establishment. On the other hand, the latency could be explained by the Hill muscle model, where the spring characteristics of the musculo-tendon units (MTUs) involve a delay between the muscle activation and the joint torque generation, needed by the tendons to be tightened. Instead of going down to the accurate but complex MTU modeling, this phenomenon could be

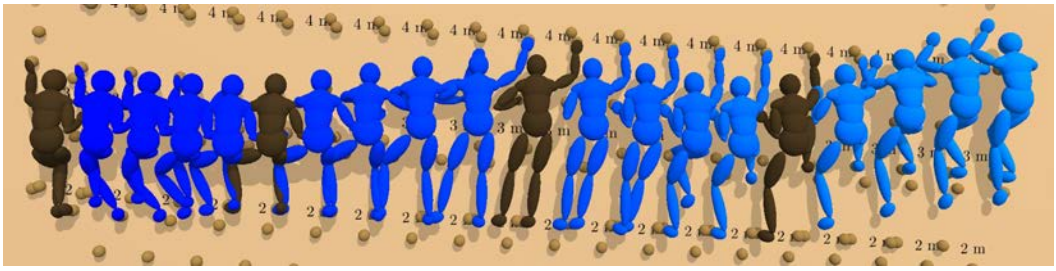
incorporated directly at joint level with a Hill-style model based on angular springs, bringing an implicit activation time.

The phases where the climber hesitates, observes the route, and takes decisions on the gestures to perform, cannot be modelized by a physical analysis. They could nevertheless be added in our physically-based approach as supplementary inputs, containing where the climber will hesitate, or will need to think about the ascent. Corresponding temporal pauses would be added at those points in the trajectory to optimize before each cost function evaluation. On the contrary, the choice of the order of the holds to take and adjustments on the climber's postures can be obtained by an optimization approach. Indeed, each of the various spatial possibilities has a particular cost, implying that a minimum exists and can be chosen for the synthesis. Moreover, the finite number of possibilities for the contact sequence makes an exhaustive test possible. Note that such an approach would be prohibitively expensive for ground motions for instance, but is affordable for climbing motions as soon as the environment kinematic restrictions are taken into account. The tractability of the optimized spatial control could also be increased by taking into account joint angular limits and the climber's self intersections to reduce the motion search space.

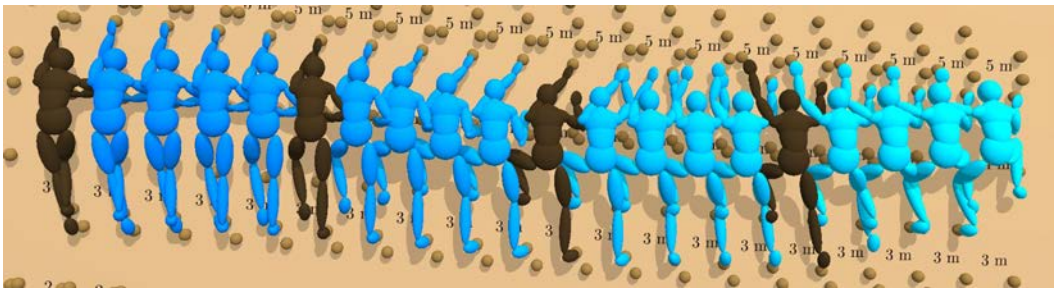
Finally the timing problem alone is enough to improve the realism of the anatomical model. The right next work in this direction is to add the fingers and their related costs. This would allow to quantify the difficulty associated to a given hold and thus to take into account in the cost function the shape of the holds. In practice, the finger cost is the dominant factor of difficulty in climbing [Guidi 1999]. Therefore, we suspect this factor to influence drastically the synthesized motions. An intermediate straightforward step toward a complete hand model is to add as input the quality of each hold. This quality can be modelized as additional weights in the hand contact costs. The other ways to improve the contact models are to add the contact normals as inputs, and to differentiate friction coefficients according to the hold characteristics, for making the difference between an handle and a flat hand hold for instance, as we have shown that these modelization issues influence the resulting optimal timing.



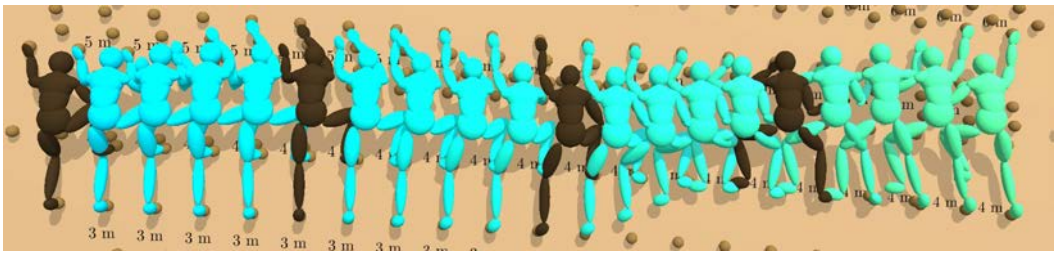
0.0 sec 0.5 sec 1.0 sec 1.5 sec



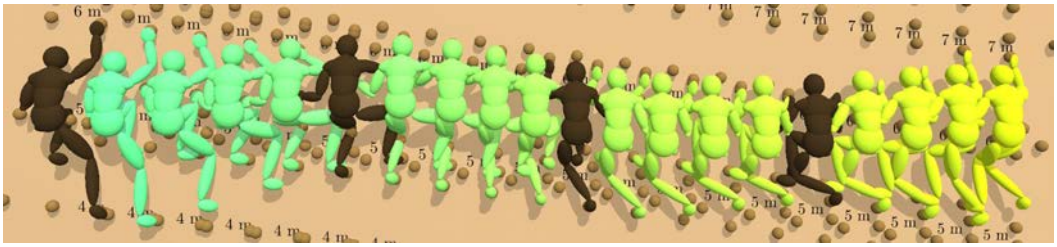
2.0 sec 2.5 sec 3.0 sec 3.5 sec



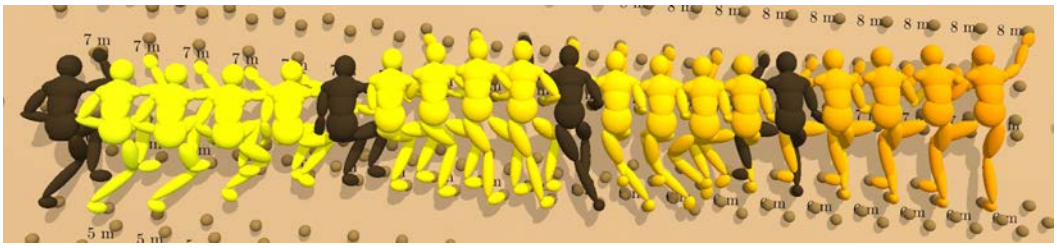
4.0 sec 4.5 sec 5.0 sec 5.5 sec



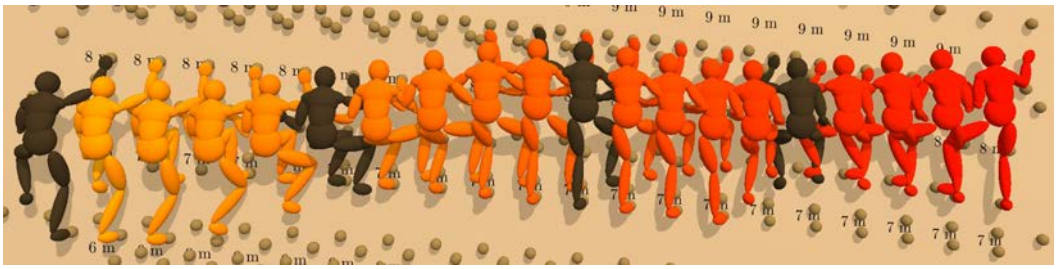
6.0 sec 6.5 sec 7.0 sec 7.5 sec



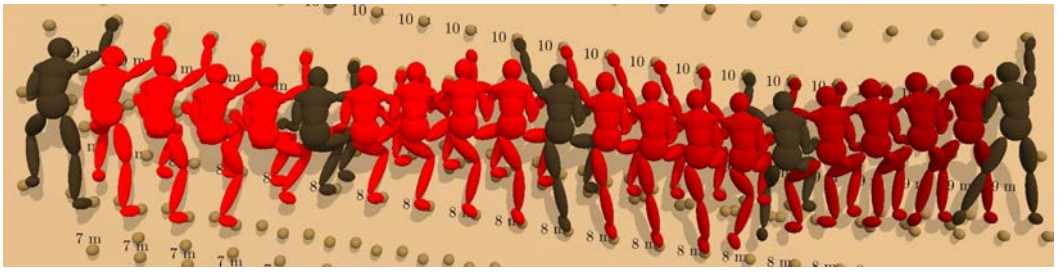
8.0 sec 8.5 sec 9.0 sec 9.5 sec



10.0 sec 10.5 sec 11.0 sec 11.5 sec



12.0 sec 12.5 sec 13.0 sec 13.5 sec



14.0 sec 14.5 sec 15.0 sec 15.5 sec 16.0 sec

Figure 4.54: *The Espace Vertical motion synthesis after thin evaluation.*

Chapter 5

Discussion and Perspectives

In this thesis, climbing motion has been studied through the capture of dynamic data, the analysis of motion by inverse dynamics and the motion synthesis by timing optimization. The contributions of this work have been the capture of calibrated 6D forces and climbing motion data with an instrumented bouldering wall, equipped with 6 force sensors and 24 infra-red cameras; the development of a joint-based sensor-free inverse dynamics algorithm for multi-contact motions, with a continuous contact transition model; and the generation of a long and realistic climbing motion sequence, from solely successive external contact sets.

The limitations concerning the data capture are first, the synchronization process between the force sensors and the MoCap system and second, the marker labelling and tracking process. A hardware synchronization between sensors and cameras would allow to go bellow 10 ms of temporal precision. This functionality is not an available feature of the devices currently used, and could necessitate some electronic original developments. Concerning the spatial precision, the current tracking black-box we used allowed us to work with only 10% of the captured data. Increasing this percentage of usable motion could be done first by correcting manually the erroneous marker labels by means of the graphic user interface (GUI) that already exists under a private licence (Optitrack, Natural Point), or second by extracting individual marker trajectories from binary MoCap intermediate files and finding a better labelling and tracking algorithm. Finally for the large scale environment, the data capture process could be complemented by adding precise environment shapes, with the use of structure from motion techniques for detailed geometric model recording. On the wall, this would allow to identify individual hold shapes in order to assess their quality and to modelize finger positioning, that would bring the possibility to evaluate finger joint costs, the key-value of the energy cost of climbing motions. Applying structure from motion on the climber itself would allow to adapt the climber's model to various body shapes and to render different climber's appearance, including its texture the day of the capture, which could be interesting for developing a texture-based environment-aware monocular motion capture algorithm for instance.

Inverse dynamics based on a least squares minimization of wrenches projected on anatomical joint axes has shown two main limitations. On one hand, the differentiation between opposite anatomical directions sharing the same axis, such as knee flexion and extension for example, can not be determined in one pass. This is problematic for including precisely joint torque limits in least squares computations because most of the main joint axes have quite different opposite torque maxima. On the other hand, for joints with at least 2 DoFs, anatomical axes obtained from Euler

angles are not orthogonal in general and can lead to axes sharing the same joint wrench component after its projection, leading to miscounted wrench costs. This inaccuracy could be resolved with a nonlinear algorithm where the wrench would not be projected, but each axis component would be added to produce the wrench needed for the motion to be perform. We suspect that this more time-consuming algorithm could be initialized with the wrenches resulting from the current fast least-square algorithm. This nonlinear joint level model would be an intermediate step between the least-square joint model currently used, and the complex and non-linear MTU-based actuation model, providing a range of compromises between accuracy and tractability ready to be used for various case studies.

Regarding motion synthesis, the main direction of improvement for our results is the implementation of the inter-movement phases, which are the instants when no hand nor foot is moving, and which contain hand or foot prehension adjustments, wall observations, hesitations, rest phases and decision making instants. Ideally for climbers, most of these events should not occur as they are energy consuming. Our work allow us to clearly identify the motions resulting from optimizing biomechanical constraints, and the unwanted behaviors due to cognitive or neurological constraints, which is the difference between our results and the observations. For graphics usage, different ways are conceivable to improve the realism of the simulated motions corresponding to the different elements constituting the inter-movement phases. The rest instants could be obtained by an advanced physiological energy model, which would assess the evolution of the available amounts of energy with respect to the different exercise energy systems (lactic/alactic, aerobic/anaerobic), and adjust the speed of the climb accordingly. The head movements and body posture adjustments for observing the wall could be obtained by simulating the field of view of the climber, who would be constrained to see the holds being taken just before their grasping. Latencies could be added for simulating the neuronal process of wide area observations. The hesitation, prehension adjustments and decision making instants can be simulated either by adding both temporal and spacial noise to the motion, or by designing a GUI for external user interventions.

For sport performance applications, improving the realism as mentioned above is not needed, except for the energy model. Indeed, a more accurate energy model would allow better optimality predictions, but all other items can be seen as elements that the climber should avoid ideally. Therefore, the major difficulty for providing to sport professionals or competitors a training tool for the gesture improvement, is finding a medium that would be accepted by this community. [Kajastila 2014] proposed a first attempt in this direction with a augmented bouldering wall on which real-time graphic information is display thanks to a video projector coupled with a Kinect camera. According to them, the use of such a wall would be better in a climbing gym as a separated wall. This drawback is probably due to the unusual equipments from a climber's point view, as well as the needed to project a massive light on the wall, disturbing the wall aspect. We suspect the community welcoming to be much higher if the deployment would be done under mobile platforms such as Android for instance, or based on small GoPro-type cameras coupled with a

laptop for outdoor sessions, as those devices are already well known and used by the potential end-users. The breakthrough needed for applying our method on such light systems is the drastic reduction of the computational cost. The most plausible way for this reduction appears to be the conversion of the non-linear timing optimization into a quadratic program by some analytical or symbolic computations. The starting point being the analytical derivation of the climber's trajectory velocity and acceleration, as initiated by the work of [McEnnan 2003].

Bibliography

[Abe 2007] Yeuhi Abe, Marco da Silva and Jovan Popović. *Multiobjective control with frictional contacts.* In Proc. SCA, pages 249–258, 2007. (Cited on page 76.)

[Al Borno 2012] Mazen Al Borno, Martin de Lasa and Aaron Hertzmann. *Trajectory Optimization for Full-Body Movements with Complex Contacts.* Transactions on Visualization and Computer Graphics, 2012. (Cited on pages 4, 75, 96 and 127.)

[Aladdin 2012] Rami Aladdin and Paul Kry. *Static pose reconstruction with an instrumented bouldering wall.* In Proceedings of the 18th ACM symposium on Virtual reality software and technology, pages 177–184. ACM, 2012. (Cited on pages 5, 32 and 36.)

[Anderson 1999] Frank C Anderson and Marcus G Pandy. *A dynamic optimization solution for vertical jumping in three dimensions.* Computer Methods in Biomechanics and Biomedical Engineering, vol. 2, no. 3, pages 201–231, 1999. (Cited on page 10.)

[Andrews 2013] S Andrews and PG Kry. *Goal directed multi-finger manipulation: Control policies and analysis.* Computers & Graphics, vol. 37, no. 7, pages 830–839, 2013. (Cited on page 5.)

[Balan 2005] A.O. Balan, L. Sigal and M.J. Black. *A Quantitative Evaluation of Video-based 3D Person Tracking.* In Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on, pages 349–356, 2005. (Cited on page 51.)

[Bhat 2002] Kiran S. Bhat, Steven M. Seitz and Jovan Popovic. *Computing the Physical Parameters of Rigid-Body Motion from Video.* In Proceedings of the 7th European Conference on Computer Vision-Part I, ECCV '02, pages 551–565, London, UK, UK, 2002. Springer-Verlag. (Cited on page 51.)

[Booth 1999] John Booth, Frank Marino, Chris Hill and Tom Gwinn. *Energy cost of sport rock climbing in elite performers.* British journal of sports medicine, vol. 33, no. 1, pages 14–18, 1999. (Cited on pages 13 and 15.)

[Bretl 2005] Tim Bretl, Jean-Claude Latombe and Stephen Rock. *Toward autonomous free-climbing robots.* In Robotics Research, pages 6–15. Springer, 2005. (Cited on page 76.)

[Brown 2013] David F. Brown, Adriano Macchietto, KangKang Yin and Victor Zordan. *Control of rotational dynamics for ground behaviors.* In Proceedings

of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '13, pages 55–61, New York, NY, USA, 2013. ACM. (Cited on page 76.)

[Brubaker 2009] Marcus A Brubaker, Leonid Sigal and David J Fleet. *Estimating contact dynamics.* In ICCV, pages 2389–2396, 2009. (Cited on pages 6, 50, 51 and 56.)

[Chu 2010] Baeksuk Chu, Kyungmo Jung, Chang-Soo Han and Daehie Hong. *A survey of climbing robots: Locomotion and adhesion.* International Journal of Precision Engineering and Manufacturing, vol. 11, no. 4, pages 633–647, 2010. (Cited on page 76.)

[Conn 2000] Andrew R Conn, Nicholas IM Gould and Philippe L Toint. Trust region methods. Siam, 2000. (Cited on page 106.)

[Coros 2011] Stelian Coros, Andrej Karpathy, Ben Jones, Lionel Reveret and Michiel Van De Panne. *Locomotion skills for simulated quadrupeds.* ACM Transaction on Graphics, vol. 30, no. 4, page 59, 2011. (Cited on page 75.)

[da Silva 2008] Marco da Silva, Yeuhi Abe and J Popović. *Simulation of Human Motion Data using Short-Horizon Model-Predictive Control.* In Computer Graphics Forum, volume 27, number 2, pages 371–380. Wiley Online Library, 2008. (Cited on page 76.)

[de Geus 2006] Bas de Geus, Seán Villanueva O'Driscoll and Romain Meeusen. *Influence of climbing style on physiological responses during indoor rock climbing on routes with the same difficulty.* European journal of applied physiology, vol. 98, no. 5, pages 489–496, 2006. (Cited on pages 14, 15 and 39.)

[de Lasa 2010] Martin de Lasa, Igor Mordatch and Aaron Hertzmann. *Feature-based locomotion controllers.* ACM Transaction on Graphics, vol. 29, no. 4, page 131, 2010. (Cited on pages 5 and 76.)

[de Moraes Bertuzzi 2007] Rômulo Cássio de Moraes Bertuzzi, Emerson Franchini, Eduardo Kokubun and Maria Augusta Peduti Dal Molin Kiss. *Energy system contributions in indoor rock climbing.* European journal of applied physiology, vol. 101, no. 3, pages 293–300, 2007. (Cited on page 10.)

[Delp 2007] S.L. Delp, F.C. Anderson, A.S. Arnold, P. Loan, A. Habib, C.T. John, E. Guendelman and D.G. Thelen. *OpenSim: Open-Source Software to Create and Analyze Dynamic Simulations of Movement.* Biomedical Engineering, IEEE Transactions on, vol. 54, no. 11, pages 1940 –1950, nov. 2007. (Cited on pages 5, 20 and 58.)

[Dupuy 1989] C Dupuy and H Ripoll. *Analyse des stratégies visuo-motrices en escalade sportive.* Revue Siences et Motricité, vol. 7, pages 19–26, 1989. (Cited on pages 10 and 139.)

[Duveau 2012] Estelle Duveau, Simon Courtemanche, Lionel Reveret and Edmond Boyer. *Cage-based Motion Recovery using Manifold Learning.* In 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on, pages 206–213. IEEE, 2012. (Cited on page 6.)

[Faloutsos 2001] Petros Faloutsos, Michiel van de Panne and Demetri Terzopoulos. *Composable controllers for physics-based character animation.* In Proc. SIGGRAPH, pages 251–260, 2001. (Cited on page 75.)

[Fang 2003] Anthony C Fang and Nancy S Pollard. *Efficient synthesis of physically valid human motion.* ACM Transaction on Graphics, vol. 22, no. 3, pages 417–426, 2003. (Cited on page 75.)

[Foddy 2008] Bennett Foddy. Qwop. http://www.foddy.net/Athletics.html, 2008. (Cited on page 76.)

[Fuss 2006] Franz Konstantin Fuss and Günther Niegl. *Instrumented climbing holds and dynamics of sport climbing.* In The Engineering of Sport 6, pages 57–62. Springer, 2006. (Cited on pages 14 and 15.)

[Fuss 2008a] Franz Konstantin Fuss and Günther Niegl. *Instrumented climbing holds and performance analysis in sport climbing.* Sports Technology, vol. 1, no. 6, pages 301–313, 2008. (Cited on pages 14 and 15.)

[Fuss 2008b] Franz Konstantin Fuss and Günther Niegl. *Quantification of the Grip Difficulty of a Climbing Hold (P142).* In The Engineering of Sport 7, pages 19–26. Springer, 2008. (Cited on pages 14 and 15.)

[Geyer 2010] H. Geyer and H. Herr. *A Muscle-Reflex Model That Encodes Principles of Legged Mechanics Produces Human Walking Dynamics and Muscle Activities.* Neural Systems and Rehabilitation Engineering, IEEE Transactions on, vol. 18, no. 3, pages 263–273, 2010. (Cited on page 76.)

[Guennebaud 2010] Gaël Guennebaud, Benoît Jacob*et al.* *Eigen v3.* http://eigen.tuxfamily.org, 2010. (Cited on page 25.)

[Guidi 1999] Olivier Guidi. *Facteurs de la performance: Les filières énergétiques en escalade.* EPS: Revue education physique et sport, no. 276, pages 15–19, 1999. (Cited on pages 10, 139 and 141.)

[Guyon 2004] Laurence Guyon and Olivier Broussouloux. Escalade et performance: préparation et entraînement. Ed. Amphora, 2004. (Cited on page 10.)

[Hague 2006] Dan M. Hague. Self-coached climber: The guide to movement, training, performance. Douglas Hunter, 2006. (Cited on page 4.)

[Hansen 2006] Nikolaus Hansen. *The CMA evolution strategy: a comparing review.* In Towards a new evolutionary computation, pages 75–102. Springer, 2006. (Cited on page 96.)

[Hazan 2007] Elad Hazan, Alexander Rakhlin and Peter L Bartlett. *Adaptive online gradient descent.* In Advances in Neural Information Processing Systems, pages 65–72, 2007. (Cited on page 106.)

[Hodgins 1995] Jessica K. Hodgins, Wayne L. Wooten, David C. Brogan and James F. O'Brien. *Animating Human Athletics.* In Proc. SIGGRAPH, pages 71–78, August 1995. (Cited on page 75.)

[Jain 2009] Sumit Jain, Yuting Ye and C Karen Liu. *Optimization-based interactive motion synthesis.* ACM Transactions on Graphics, vol. 28, no. 1, page 10, 2009. (Cited on pages 5 and 76.)

[Jain 2011] Sumit Jain and C Karen Liu. *Controlling physics-based characters using soft contacts.* In ACM Transaction on Graphics (SIGGRAPH Asia), volume 30, number 6, page 163, 2011. (Cited on page 76.)

[Kajastila 2014] R. Kajastila and P. Hämäläinen. *Augmented Climbing: Interacting With Projected Graphics on a Climbing Wall.* CHI Extended Abstracts on Human Factors in Computing Systems, 2014. (Cited on page 146.)

[Kry 2006] Paul G. Kry and Dinesh K. Pai. *Interaction capture and synthesis.* ACM Trans. Graph., vol. 25, no. 3, pages 872–880, 2006. (Cited on page 5.)

[Kry 2009] Paul G Kry, Lionel Revéret, François Faure and M-P Cani. *Modal locomotion: Animating virtual characters with natural vibrations.* Computer Graphics Forum, vol. 28, no. 2, pages 289–298, 2009. (Cited on page 4.)

[Ladha 2013] Cassim Ladha, Nils Y Hammerla, Patrick Olivier and Thomas Plötz. *ClimbAX: skill assessment for climbing enthusiasts.* In Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing, pages 235–244. ACM, 2013. (Cited on pages 14 and 15.)

[Laszlo 2000] Joseph Laszlo, Michiel van de Panne and Eugene Fiume. *Interactive Control for Physically-based Animation.* In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '00, pages 201–208, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. (Cited on page 76.)

[Lechner 2013] B Lechner, I Filzwieser, M Lieschnegg and P Sammer. *A Climbing Hold With an Integrated Three Dimensional Force Measurement and Wireless Data Acquisition.* International Journal on Smart Sensing and Intelligent Systems, vol. 6, no. 5, 2013. (Cited on pages 14 and 15.)

[Lee 2006] Sung-Hee Lee and Demetri Terzopoulos. *Heads up!: biomechanical modeling and neuromuscular control of the neck.* In ACM Transactions on Graphics, volume 25, number 3, pages 1188–1198. ACM, 2006. (Cited on page 76.)

[Lee 2009] Sung-Hee Lee, Eftychios Sifakis and Demetri Terzopoulos. *Comprehensive biomechanical modeling and simulation of the upper body.* ACM Transactions on Graphics, vol. 28, no. 4, page 99, 2009. (Cited on page 76.)

[Linder 2005] Stephen Paul Linder, Edward Wei and Alexander Clay. *Robotic rock climbing using computer vision and force feedback.* In Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, pages 4685–4690. IEEE, 2005. (Cited on page 76.)

[Liu 2005] C Karen Liu, Aaron Hertzmann and Zoran Popović. *Learning physics-based motion style with nonlinear inverse optimization.* In ACM Transaction on Graphics, volume 24, number 3, pages 1071–1081, 2005. (Cited on pages 52, 75 and 117.)

[Liu 2009] C Karen Liu. *Dextrous manipulation from a grasping pose.* ACM Transaction on Graphics, vol. 28, no. 3, page 59, 2009. (Cited on page 5.)

[Liu 2010] Libin Liu, KangKang Yin, Michiel van de Panne, Tianjia Shao and Weiwei Xu. *Sampling-based contact-rich motion control.* ACM Transactions on Graphics, vol. 29, no. 4, page 128, 2010. (Cited on page 76.)

[Liu 2013] Karen Liu, A. Huamán Quispe, Jeongseok Lee, Mike Stilman, Jonathan Martin, Pushkar Kolhe, Saul Reynolds-Haertle and Tobias Kunz. Dart: Dynamic animation and robotics toolkit. http://dartsim.github.io, 2013. (Cited on page 25.)

[Macchietto 2009] Adriano Macchietto, Victor Zordan and Christian R Shelton. *Momentum control for balance.* In ACM Transactions on Graphics, volume 28, number 3, page 80. ACM, 2009. (Cited on pages 4, 76 and 125.)

[Magiera 2013] Artur Magiera, Robert Roczniok, Adam Maszczyk, Miłosz Czuba, Joanna Kantyka and Piotr Kurek. *The Structure of Performance of a Sport Rock Climber.* Journal of human kinetics, vol. 36, no. 1, pages 107–117, 2013. (Cited on pages 13, 14 and 15.)

[McCann 2006] James McCann, Nancy S Pollard and Siddhartha Srinivasa. *Physics-based motion retiming.* In Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 205–214. Eurographics Association, 2006. (Cited on page 77.)

[McEnnan 2003] James McEnnan. Quaternion cubic spline. http://qspline.sourceforge.net/qspline.pdf, May 2003. (Cited on pages 82 and 147.)

[Mermier 2000] Christine M Mermier, Jeffrey M Janot, Daryl L Parker and Jacob G Swan. *Physiological and anthropometric determinants of sport climbing performance.* British Journal of Sports Medicine, vol. 34, no. 5, pages 359–365, 2000. (Cited on pages 14 and 15.)

[Moeslund 2001] Thomas B Moeslund and Erik Granum. *A survey of computer vision-based human motion capture.* Computer Vision and Image Understanding, vol. 81, no. 3, pages 231–268, 2001. (Cited on page 6.)

[Mordatch 2010] Igor Mordatch, Martin De Lasa and Aaron Hertzmann. *Robust physics-based locomotion using low-dimensional planning.* ACM Transactions on Graphics, vol. 29, no. 4, page 71, 2010. (Cited on page 75.)

[Mordatch 2012] Igor Mordatch, Emanuel Todorov and Zoran Popović. *Discovery of complex behaviors through contact-invariant optimization.* ACM Transactions on Graphics, vol. 31, no. 4, page 43, 2012. (Cited on pages 4, 52, 56 and 75.)

[Mordatch 2013] Igor Mordatch, Jack M. Wang, Emanuel Todorov and Vladlen Koltun. *Animating Human Lower Limbs Using Contact-Invariant Optimization.* ACM Transactions on Graphics, vol. 32, no. 6, 2013. (Cited on pages i, iii, 10, 83, 84 and 96.)

[Murray 1994] Richard M Murray and S Shankar Sastry. A mathematical introduction to robotic manipulation. CRC press, 1994. (Cited on pages 25 and 53.)

[Noé 2001] Frédéric Noé, F Quaine and L Martin. *Influence of steep gradient supporting walls in rock climbing: biomechanical analysis.* Gait & posture, vol. 13, no. 2, pages 86–94, 2001. (Cited on page 5.)

[Nunes 2012] Rubens F Nunes, Paul G Kry and Victor B Zordan. *Using natural vibrations to guide control for locomotion.* In Proc. I3D, pages 87–94. ACM, 2012. (Cited on page 4.)

[Ondra 2013] Adam Ondra and Peter Pavlíček. Change, the story of the first 9b+. http://www.youtube.com/watch?v=mgz7ybI1frc, 2013. (Cited on page 39.)

[Pansiot 2008] Julien Pansiot, Rachel C King, Douglas G McIlwraith, Benny PL Lo and Guang-Zhong Yang. *ClimBSN: Climber performance monitoring with BSN.* In Medical Devices and Biosensors, 2008. ISSS-MDBS 2008. 5th International Summer School and Symposium on, pages 33–36. IEEE, 2008. (Cited on pages 14 and 15.)

[Popović 1999] Zoran Popović and Andrew Witkin. *Physically based motion transformation.* In SIGGRAPH, pages 11–20, 1999. (Cited on page 75.)

[Quaine 1999] F Quaine and L Martin. *A biomechanical study of equilibrium in sport rock climbing.* Gait & Posture, vol. 10, no. 3, pages 233–239, 1999. (Cited on pages 5, 13 and 15.)

[Quaine 2011] Franck Quaine, Laurent Vigouroux, Florent Paclet, Floren Colloud*et al. The thumb during the crimp grip.* International journal of sports medicine, vol. 32, no. 1, page 49, 2011. (Cited on page 5.)

[Raibert 1991] Marc H Raibert and Jessica K Hodgins. *Animation of dynamic legged locomotion.* In Proc. SIGGRAPH, volume 25, number 4, pages 349–358. ACM, 1991. (Cited on page 75.)

[Ren 2010] Cheng Ren, Liming Zhao and Alla Safonova. *Human Motion Synthesis with Optimization-based Graphs.* Computer Graphics Forum, vol. 29, no. 2, pages 545–554, 2010. (Cited on page 75.)

[Righetti 2013] Ludovic Righetti, Jonas Buchli, Michael Mistry, Mrinal Kalakrishnan and Stefan Schaal. *Optimal distribution of contact forces with inverse dynamics control.* The International Journal of Robotics Research, 2013. (Cited on pages 51, 52 and 54.)

[Robert 2013] T Robert, J Causse and G Monnier. *Estimation of external contact loads using an inverse dynamics and optimization approach: General method and application to sit-to-stand maneuvers.* Journal of biomechanics, vol. 46, no. 13, pages 2220–2227, 2013. (Cited on pages 51, 52, 57, 58, 62, 65, 68 and 70.)

[Rougier 1992] P Rougier and J.P. Blanchi. *Mesure de la force maximale volontaire à partir d'une posture quadrupodale en escalade: influence du niveau d'expertise.* Science & Sports, vol. 7, no. 1, pages 19 – 25, 1992. (Cited on pages 13 and 15.)

[Safonova 2008] Alla Safonova and JessicaK. Hodgins. *Synthesizing Human Motion from Intuitive Constraints.* In Dimitri Plemenos and Georgios Miaoulis, editeurs, Artificial Intelligence Techniques for Computer Graphics, volume 159 of *Studies in Computational Intelligence*, pages 15–39. Springer Berlin Heidelberg, 2008. (Cited on page 75.)

[Salzmann 2011] M. Salzmann and R. Urtasun. *Physically-based motion models for 3D tracking: A convex formulation.* In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 2064–2071, 2011. (Cited on pages 6 and 51.)

[Schmid 2007] Thomas Schmid, Roy Shea, Jonathan Friedman and Mani B Srivastava. *Movement Analysis in Rock-Climbers.* In Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on, pages 567–568. IEEE, 2007. (Cited on pages 14 and 15.)

[Schoeffl 2004] V Schoeffl, S Klee and W Strecker. *Evaluation of physiological standard pressures of the forearm flexor muscles during sport specific ergometry in sport climbers.* British journal of sports medicine, vol. 38, no. 4, pages 422–425, 2004. (Cited on pages 14 and 15.)

[Schweizer 2001] Andreas Schweizer. *Biomechanical properties of the crimp grip position in rock climbers.* Journal of biomechanics, vol. 34, no. 2, pages 217–223, 2001. (Cited on page 5.)

[Schöffl 2007] VR Schöffl and I Schöffl. *Finger pain in rock climbers: reaching the right differential diagnosis and therapy.* The Journal of sports medicine and physical fitness, vol. 47, no. 1, pages 70–78, 2007. (Cited on page 5.)

[Sheel 2004] AW Sheel. *Physiology of sport rock climbing.* British journal of sports medicine, vol. 38, no. 3, pages 355–359, 2004. (Cited on pages 9, 14 and 15.)

[Sibella 2007] Federica Sibella, Iuri Frosio, F Schena and NA Borghese. *3D analysis of the body center of mass in rock climbing.* Human movement science, vol. 26, no. 6, pages 841–852, 2007. (Cited on pages 14 and 15.)

[Stewart 1996] David Stewart and J. C. Trinkle. *An implicit time-stepping scheme for rigid body dynamics with Coulomb friction.* International Journal of Numerical Methods in Engineering, vol. 39, pages 2673–2691, 1996. (Cited on page 54.)

[Testa 1999] M Testa, L Martin and B Debu. *Effects of the type of holds and movement amplitude on postural control associated with a climbing task.* Gait & Posture, vol. 9, no. 1, pages 57–64, 1999. (Cited on pages 13 and 15.)

[Testa 2003] M Testa, L Martin and B Debû. *3D analysis of posturo-kinetic coordination associated with a climbing task in children and teenagers.* Neuroscience letters, vol. 336, no. 1, pages 45–49, 2003. (Cited on pages 13 and 15.)

[Urtasun 2006] R. Urtasun, D.J. Fleet and P. Fua. *3D People Tracking with Gaussian Process Dynamical Models.* In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, volume 1, pages 238–245, 2006. (Cited on page 51.)

[Vondrak 2012a] M. Vondrak, L. Sigal and O.C. Jenkins. *Dynamical Simulation Priors for Human Motion Tracking.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 35, no. 1, pages 52–65, 2012. (Cited on pages 6 and 52.)

[Vondrak 2012b] Marek Vondrak, Leonid Sigal, Jessica Hodgins and Odest Jenkins. *Video-based 3D motion capture through biped control.* ACM Trans. Graph., vol. 31, no. 4, pages 27:1–27:12, July 2012. (Cited on pages 50, 52 and 75.)

[Wampler 2009] Kevin Wampler and Zoran Popović. *Optimal gait and form for animal locomotion.* In ACM Transactions on Graphics, volume 28, number 3, page 60. ACM, 2009. (Cited on page 75.)

[Wang 2010] Jack M Wang, David J Fleet and Aaron Hertzmann. *Optimizing walking controllers for uncertain inputs and environments.* ACM Transactions on Graphics, vol. 29, no. 4, page 73, 2010. (Cited on page 75.)

[Wang 2012] Jack M Wang, Samuel R Hamner, Scott L Delp and Vladlen Koltun. *Optimizing locomotion controllers using biologically-based actuators and objectives.* ACM Transactions on Graphics (TOG), vol. 31, no. 4, page 25, 2012. (Cited on pages 75 and 76.)

[Watts 2004] Phillip B Watts. *Physiology of difficult rock climbing.* European journal of applied physiology, vol. 91, no. 4, pages 361–372, 2004. (Cited on pages 14 and 15.)

[Wei 2010] Xiaolin Wei and Jinxiang Chai. *VideoMocap: modeling physically realistic human motion from monocular video sequences.* ACM Trans. Graph., vol. 29, no. 4, pages 42:1–42:10, July 2010. (Cited on pages 52, 54, 75 and 83.)

[Witkin 1988] Andrew Witkin and Michael Kass. *Spacetime constraints.* In Proc. SIGGRAPH, volume 22, number 4, pages 159–168, 1988. (Cited on page 75.)

[Wu 1995] G. Wu and P. R. Cavanagh. *ISB recommendations for standardization in the reporting of kinematic data.* Journal of biomechanics, vol. 28, no. 10, pages 1257–1261, 1995. (Cited on page 58.)

[Wu 2002] G. Wu, S. Siegler, P. Allard, C. Kirtley, A. Leardini, D. Rosenbaum, M. Whittle, D.D. D'Lima, L. Cristofolini, H. Witte, O. Schmid and I. Stokes. *ISB recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion–part I: ankle, hip, and spine.* Journal of biomechanics, vol. 35, pages 543–548, 2002. (Cited on page 58.)

[Wu 2005] G. Wu, F.C.T. van der Helm, H.E.J. (DirkJan) Veeger, M. Makhsous, P. Van Roy, C. Anglin, J. Nagels, A.R. Karduna, K. McQuade, X. Wang, F.W. Werner and B. Buchholz. *ISB recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion–part II: shoulder, elbow, wrist and hand.* Journal of biomechanics, vol. 38, pages 981–992, 2005. (Cited on page 58.)

[Wu 2011] Changchang Wu. Visualsfm: A visual structure from motion system. http://www.cs.washington.edu/homes/ccwu/vsfm, 2011. (Cited on pages 46 and 48.)

[Yin 2007] KangKang Yin, Kevin Loken and Michiel van de Panne. *SIMBICON: simple biped locomotion control.* In ACM Transaction on Graphics, volume 26, number 3, page 105, 2007. (Cited on pages 4 and 75.)

[Zordan 2006] Victor B Zordan, Bhrigu Celly, Bill Chiu and Paul C DiLorenzo. *Breathe easy: Model and control of human respiration for computer animation.* Graphical Models, vol. 68, no. 2, pages 113–132, 2006. (Cited on page 76.)