



HAL
open science

From Biomechanics to Robotics

Galo Maldonado, Philippe Souères, Bruno Watier

► **To cite this version:**

Galo Maldonado, Philippe Souères, Bruno Watier. From Biomechanics to Robotics. Springer. Biomechanics of Anthropomorphic Systems, 2018, Star 124. hal-02002519

HAL Id: hal-02002519

<https://laas.hal.science/hal-02002519>

Submitted on 31 Jan 2019

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.

From Biomechanics to Robotics

Galo Maldonado, Philippe Souères, Bruno Watier

Abstract How does the central nervous system select and coordinate different degrees of freedom to execute a given movement? The difficulty is to choose one specific motor command among an infinite number of possible ones. If some invariants of movement can be identified, there exists however considerable variability showing that motor control favors rather an envelope of possible movements than a strong stereotypy. However, the central nervous system is able to find extremely fast solutions to the problem of muscular and kinematic redundancy by producing stable and precise movements. To date, no computational model has made it possible to develop a movement generation algorithm with a performance comparable to that of humans in terms of speed, accuracy, robustness and adaptability. One of the reasons is certainly that correct criteria for the synthesis of movement as a function of the task have not yet been identified and used for motion generation. In this chapter we propose to study highly dynamic human movement taking into account its variability, making the choice to consider performance biomechanical variables (tasks) for generating motions and involving whole-body articulations.

1 Introduction

Humanoid robotics is growing towards agile, robust, and powerful robots able to interact with environments in which humans might be included [49]. To control these robots, motion generation algorithms are being improved to increase the robot sta-

Galo Maldonado
LAAS-CNRS, Université de Toulouse, CNRS, UPS, Toulouse, France e-mail: gmaldona@laas.fr

Philippe Souères
LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France e-mail: soueres@laas.fr

Bruno Watier
LAAS-CNRS, Université de Toulouse, CNRS, UPS, Toulouse, France e-mail: bruno.watier@univ-tlse3.fr

bility, robustness, and efficiency [4, 37]. Though a large part of control frameworks are purely computational, some of them take inspiration from humans. This is not an arbitrary choice. Understanding human motion could provide the information needed to improve the design and control of humanoid robots. Modeling human motion has led to applications in the movie and gaming industries for realistic animation of characters. In the robotic industry, collaborative robots have emerged over the past years with the objective to assist humans in their daily live activities. Examples include assistance robots for elderly people, exoskeletons to extend human physical capabilities, brain-machine interfaces connected to robotic arms, virtual reality training systems for surgery, or controlled prostheses.

Humans and robots share some important characteristics such as under-actuation and redundancy. Under-actuation means that the number of actuated degrees of freedom "DoF" of the system is lower than the total number of available DoF. The under-actuated part are 6 DoF that define the body pose: position and orientation. In robotics these 6 parameters describe the pose of a reference frame usually called root frame. Redundancy expresses the fact that the system has more DoF than necessary to achieve a given task. The difficulty is then to select a motor command among an infinite of possible ones. In humans, redundancy appears at the level of neurons, muscles, and joints. In robots, it requires the design of motion generation algorithms to select the best solution for controlling a movement. In biomechanics, redundancy is studied by computing the kinematics and dynamics of human motion which allows to explain motor control strategies in terms of performance variables (tasks).

In this chapter, a set of tools and methods in biomechanics and robotics and a case study of human-inspired motion generation is presented. The chapter is organized as follows. Section 2 provides a brief summary of the methods used in biomechanics to study human motion. A summary of the robotic framework used to generate whole-body motion is then given in section 3. Section 4 presents a case study of a human dynamic motion which was analyzed through biomechanical methods and generated using the robotic framework. Lastly, a summary of the method used to generate human-inspired motion from biomechanics to robotics is given in section 5.

2 Biomechanics background

Biomechanics aims at applying the physical laws of mechanics to study human motion in order to provide a better understanding of the mechanical strategies used by the central nervous system (CNS) to coordinate motions. This information can be used to generate human-inspired motions with humanoid robots or anthropomorphic systems such as animation avatars. This section provides a summary of the mathematical and computational background needed to study the biomechanics of the human motion.

2.1 Motion Capture

Motion capture "MoCap" is a technology that allows motion to be recorded online. In biomechanics, it is used to record the movements of humans and animals for quantitative analyses. The motion is tracked using cameras that record the 2D positions of markers placed on the body. When more than one camera is available, the 3D marker coordinates can be computed by the MoCap system. Two types of markers can be used depending on the motion capture system: active and passive markers.

To reconstruct the motion, active systems use infra-red active markers with light-emitting diodes stimulated in a predefined sequence. On the other hand, passive systems use markers covered by a reflective tape which are lighted by infra-red cameras. Coordinate data are calculated by the system with respect to a laboratory fixed reference frame. Data are then processed to obtain kinematic variables describing segment or joint movements. A motion capture system might also synchronize other information such as reaction forces through force sensors or muscular activity through electromyography (EMG) sensors. In order to communicate these data, an experimental protocol must be carefully described and followed. The recording protocol should take into account recommendations for reporting biomechanics data based on international standards. For example, kinematic data are collected and reported based on recommendations from the International Society of Biomechanics (ISB) [53, 54] and surface muscular activity is recorded and communicated based on the recommendations from the European project of Surface ElectroMyoGraphy for the Non-Invasive Assessment of Muscles (SENIAM).

2.2 Human data processing

Recorded motion data have to be reconstructed and processed. Data reconstruction consists in labeling markers and interpolating their position to reconstruct their trajectories. Next, data are filtered to remove noise from the recorded and reconstructed signals. For filtering the reconstructed data, a low-pass Butterworth digital filter applied in a zero-phase is commonly used in biomechanics. In order to filter force and marker trajectories, the same cut-off frequency is chosen to avoid inconsistencies with inverse dynamics computations (inverse dynamics of human motion will be presented in 2.5) [31, 38]. To select the cut-off frequency for filtering a signal, two types of analyses are commonly performed: power spectral analysis and/or residual analysis [52]. Power spectral analysis is a technique in which the power of the recorded signals can be studied in the frequency domain. Based on this analysis, a decision can be made to define which frequencies are to be accepted or rejected from the signals. Residual analysis [52] is used to evaluate noise by comparing the difference between the unfiltered signals and the signals filtered at different cut-off frequencies. Let N_f be the number of i sampled frames in a signal, the residual ε at a given cut-off frequency f_c is calculated as follows:

$$\varepsilon(f_c) = \frac{1}{N_f} \sum_{i=1}^{N_f} (s_i - \hat{s}_i(f_c))^2, \quad (1)$$

where N_f is the number of recorded motion frames, s_i is the raw signal, \hat{s}_i is the filtered signal with cut-off frequency f_c . Eq. (1) is calculated recursively for a set of given cut-off frequencies. The set of cut-off frequencies is arbitrary but should be chosen so that the frequencies of the signal can be represented. Residuals are later displayed as a function of these cut-off frequencies to evaluate the contained information. Criteria related to the choice of the cutoff frequency based on residual analysis can be found in [52].

After data have been filtered, joint centers of rotation can be estimated from markers' trajectories. Three methods can be used to this end: virtual models (e.g. in OpenSim software [8]), regression tables (e.g. [10]) and/or functional methods (e.g. [14]). In general, the center of rotation of ball-modeled joints such as the shoulder and the hip are computed by using functional methods. The SCoRe method provides a good estimation [14] and can be computed as:

$$\min_{\mathbf{c}_1, \mathbf{c}_2} \sum_{i=1}^{N_f} \|\mathbf{R}_{1_i} \mathbf{c}_1 + \mathbf{t}_{1_i} - (\mathbf{R}_{2_i} \mathbf{c}_2 + \mathbf{t}_{2_i})\|^2, \quad (2)$$

where $\mathbf{c}_1 \in \mathbb{R}^3$ and $\mathbf{c}_2 \in \mathbb{R}^3$ are the centers of rotation in the body segment coordinate system, $(\mathbf{R}_{1_i}, \mathbf{t}_{1_i})$ and $(\mathbf{R}_{2_i}, \mathbf{t}_{2_i})$ are respectively the transformation matrices of body 1 and body 2 from segment (local) to world (global) coordinates. The solution to this minimization problem provides two trajectories \mathbf{c}_1 and \mathbf{c}_2 for the same joint and thus the mean value of both can be used as an estimate. Functional methods can also be used to determine the optimal axes of joint rotations (e.g. for calculating the knee axes [15]). Other joints center can be computed based on regression tables or based on a virtual model (which will be introduced in subsection 2.3.1).

2.3 Scaling of human anthropometry

The human body is composed by hundreds of muscles and bones. In order to calculate the kinematics and dynamics, a physical model representing the skeletal or musculoskeletal system is needed. The model is simplified by the assumption that the human body can be described by a collection of rigid bodies (segments) representing bones or a combination of them (for example when modeling the torso or the foot segments). In spite of this simplification, this model is usually called skeletal model and the same convention will be used in this chapter. In the case of muscle studies, a more complex model of the system is needed: musculoskeletal models.

The scaling of anthropometry is used to estimate the properties of the human body segments such as segment lengths, inertia matrices, and center of mass positions. To estimate human anthropometry, cadaver studies [10], mathematical modeling [24], scanning and imaging techniques [55, 13], kinematic measurements [23],

or motion capture based identification with kinetic measurements [50], can be used. Commonly, experimental marker data from static trials are used to scale the anthropometry of the recorded participant using the regression equations provided by cadaver studies or by fitting these data with a virtual skeletal model (Fig. 1). Further details about the virtual skeletal model are given in section 2.3.1 while the data fitting technique is presented later in subsection 2.4.

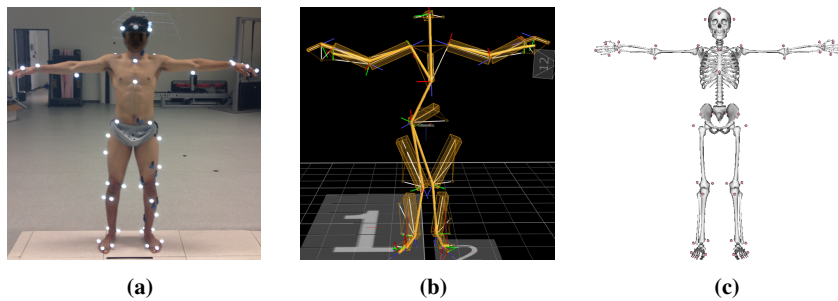


Fig. 1: In (a), a human with passive markers on his body. In (b), the model created for motion reconstruction with the Vicon Nexus software. In (c), a virtual skeleton based on OpenSim models which can be used for scaling human anthropometry.

2.3.1 Human skeletal model

A skeletal model is an effective tool for visualizing and analyzing human motion. When building a virtual 3D-model, the following information has to be considered:

- **The model** contains the description of the kinematic chain including joint types and joint ranges of motion.
- **Segment data** provide specifications of the physical characteristics of the body segments such as masses, inertia matrices, and positions of the center of mass of each segment.
- **Virtual markers** contain the positions of the markers placed on the model according to the experimental protocol. Markers are normally placed in accordance with the International Society of Biomechanics (ISB) standards with a minimum of 3 virtual markers per segment for a 3D analysis [53, 54].
- **The visual elements** are the 3D meshes that will be displayed, which can be created using a 3D software. These data are useful for visualization purposes but do not interfere with mathematical calculations.

2.4 Inverse kinematics of human motion

Inverse kinematics of human motion consists in converting experimental marker positions into joint angles by minimizing the error coming from the reconstructed motion and from the soft tissue artifact "STA". STA corresponds to the relative motion of soft tissues with regards to the underlying bones. Two classes of methods are commonly used in the literature to calculate 3D angles: segment optimization [7] and global optimization [35]. In segment optimization, a reference frame is positioned and oriented on each modeled body segment according to the ISB recommendations. Angles are then calculated following Euler angle sequences by computing the optimal bone pose from a marker cluster. Let N_f be the total number of frames recorded in a given motion and let $N_{m,s}$ be the number of markers in segment s (at least 3 markers are needed to reconstruct the 3D coordinates of a segment). At a given frame, let $x_{s,prev}$ represent the 3D position of marker s in the previous frame and $x_{s,next}$ the 3D position of marker s in the next frame. Segment optimization methods are based on the following least-square problem:

$$\min_{\mathbf{d}, \mathbf{R}} \sum_{i=1}^{N_{m,s}} \|\mathbf{R}x_{i,prev} + \mathbf{d} - x_{i,next}\|^2, \quad (3)$$

where \mathbf{R} and \mathbf{d} are the rotation matrix and the translation vector respectively, which map coordinates $x_{i,prev}$ to $x_{i,next}$ so that $(\mathbf{R}, \mathbf{d}) : x_{i,prev} \mapsto x_{i,next}$. Global optimization methods, usually referred to as "Inverse Kinematics", can also be used to calculate joint angles and constitute a promising methodology [12]. Global optimization makes it possible to add physically realistic joint constraints while taking into account the whole kinematic chain structure, and joint ranges of motion. Inverse kinematics solves the following problem:

$$\begin{aligned} \min_{\mathbf{q}} \quad & \sum_{i=1}^{N_m} w_i \|x_i^{exp} - \mathbf{x}_i(\mathbf{q})\|^2 \\ \text{s.t.} \quad & \underline{\mathbf{q}} \leq \mathbf{q} \leq \bar{\mathbf{q}}, \end{aligned} \quad (4)$$

where N_m is the total number of markers, \mathbf{q} are the generalized coordinates, $\underline{\mathbf{q}}$ and $\bar{\mathbf{q}}$ are minimum and maximum ranges of motion of the joint coordinates \mathbf{q} , x_i^{exp} is the experimental marker position of the i^{th} marker, $\mathbf{x}_i(\mathbf{q})$ is the corresponding virtual model marker position, and w_i is the marker weight, which specifies how strongly marker error should be minimized.

2.5 Inverse dynamics of human motion

Inverse dynamics aims at determining internal forces and joint torques that generate a given motion. To this end, the body model described before (skeletal model), the

movement kinematics and the measured external forces produced if there is contact with the environment can be used. Three formalism are available to compute inverse dynamics: Hamiltonian, Euler-Lagrange and Newton-Euler. In particular, Newton-Euler formulation expresses dynamic equations for each link and performs calculations recursively by propagating reaction forces and applying Newton third law of motion (principle of action and reaction). Euler's first and second equations of motion form the so called Newton-Euler equations. The first equation states that the sum of external forces equals the variation of linear momentum:

$$\sum \mathbf{f}_{ext} = \frac{d}{dt}(m\mathbf{v}) = m\mathbf{a}, \quad (5)$$

where m is the mass of the body and \mathbf{a} its the center of mass acceleration. The second equation states that the sum of external torques equals the variation of angular momentum at the center of mass (CoM):

$$\sum \boldsymbol{\tau}_{ext} = \frac{d}{dt}(I_G \boldsymbol{\omega}) = I_G \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times I_G \boldsymbol{\omega}, \quad (6)$$

where I_G is the inertia, $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\omega}}$ are respectively the body inertia, angular velocity and angular acceleration expressed at the CoM. Eq. (5) and Eq. (6) are commonly propagated at each joint using a bottom-up approach. This method can be computed numerically though the application of the recursive formulation of Newton-Euler equations [32]. Four inverse dynamics methods have been proposed in the literature based on vectors and Euler angles, wrenches and quaternions, homogeneous matrices, or generalized coordinates and forces [9].

3 Whole-body motion generation of anthropomorphic systems

Whole-body motion generation of anthropomorphic systems, such as humanoid robots, requires to model the system dynamics, in order to create stable and feasible movements, and to solve redundancy. In this section, the task function approach [44, 39] and the formalism of poly-articulated systems are recalled. The inverse kinematics and inverse dynamics formulation for solving the equations of motion in robotics are briefly introduced. Finally, a hierarchical task controller which aims at solving strict hierarchy problems is also reviewed.

3.1 Dynamic Model

The dynamic model of humanoid robots can be written by using the Euler-Lagrange equation. The robotic system is modeled as an under-actuated kinematic-tree chain composed of rigid bodies with a free-floating base (also called root frame) subject

to external contact forces as follows:

$$M(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) = S^T \boldsymbol{\tau}_{int} + \sum_{k=1}^K J_k^T(\mathbf{q}) \boldsymbol{\lambda}_k, \quad (7)$$

where $M(\mathbf{q})$ is the mass matrix, $\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}})$ contains gravitational, centrifugal and Coriolis forces, $S = \begin{bmatrix} \mathbf{0}_{n \times 6} & \mathbf{I}_{n \times n} \end{bmatrix}$ is a matrix that selects the joint torques $\boldsymbol{\tau}_{int}$ of the actuated part of Eq. (7), $J_k(\mathbf{q})$ is the Jacobian matrix of the k^{th} external contact and $\boldsymbol{\lambda}_k = [\mathbf{f}_{ext_k} \quad \boldsymbol{\tau}_{ext_k}]^T$ is the vector of the external forces and torques induced by the k^{th} contact.

3.2 Task formalism for motion generation

Let n be the number of DoF of the system, Q the configuration space formed by $n - 6$ joints plus 6 parameters of pose (position and orientation) of the system root frame and $\mathbf{q}(t) \in Q$ the configuration vector at time t . Dependencies on time will be dropped for notation convenience when necessary. Let m be the dimension of the task space also called operational space [30]. A task function $\mathbf{e}(\mathbf{q}) \in \mathbb{R}^m$ comes down to an output error function whose regulation to zero corresponds to the execution of the task. For instance, a pointing task can be defined by the task function $\mathbf{e}(\mathbf{q}) = \mathbf{hand}(\mathbf{q}) - \mathbf{hand}_{target}$, which describes the gap between the current hand position $\mathbf{hand}(\mathbf{q})$ when the body is at configuration \mathbf{q} and the expected hand position \mathbf{hand}_{target} . In order to compute how the task function varies with respect to the body configuration, roboticists use the so called task Jacobian $J_e = \frac{\partial \mathbf{e}_i}{\partial \mathbf{q}_j}$. The derivative of the task function $\mathbf{e}(\mathbf{q})$ with respect to time gives the first order kinematics relation:

$$\dot{\mathbf{e}}(\mathbf{q}, \dot{\mathbf{q}}) = J_e(\mathbf{q}) \dot{\mathbf{q}}. \quad (8)$$

The execution of the task can be regulated with a control law by specifying a reference behavior of the task $\dot{\mathbf{e}}^*$, for example with a proportional derivative (PD) control law. The gains of the proportional derivative task can be tuned to obtain different reference behaviors such as exponential decays or adaptive gains of the task. Exponential decay control laws of the form $\dot{\mathbf{e}}^* = -\lambda \mathbf{e}$ are commonly used in robotics to make the task function converge quickly to a desired value. If reference behaviors can be extracted from humans in terms of performance variables, they can be used instead to design human-inspired decay dynamics for the task function. For example, the minimum jerk criterion observed in human motions [19], has been used in the control of reaching tasks [27].

The task function approach can be extended to higher order derivatives for studying dynamic behaviors. By differentiating Eq. (8), the second order kinematic relation is obtained:

$$\ddot{\mathbf{e}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = J_e(\mathbf{q}) \ddot{\mathbf{q}} + \dot{J}_e(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}, \quad (9)$$

where $\dot{J}_e \dot{\mathbf{q}}$ can be considered as a drift of the task.

The same formalism can be used to express other types of tasks. For example, it can be used to formulate tasks for controlling the behavior of the center of mass "CoM". Let n_s represent the number of segments of the system, $m_{s_i} \in \mathbb{R}$ the mass of the i^{th} segment, and $\mathbf{c}_i \in \mathbb{R}^3$ its CoM position. The CoM of the system $\mathbf{c} \in \mathbb{R}^3$ can be computed as follows:

$$\mathbf{c}(\mathbf{q}) = \frac{\sum_{i=1}^{n_s} m_{s_i} \mathbf{c}_i}{\sum_{i=1}^{n_s} m_{s_i}}. \quad (10)$$

The CoM task can then be expressed as $\mathbf{e}(\mathbf{q}) = \mathbf{c}(\mathbf{q}) - \mathbf{c}^*$, where \mathbf{c}^* is the expected CoM position.

Dynamic tasks can also be expressed in terms of momenta in order to consider inertial effects. Momenta tasks can be computed based on the centroidal dynamics: the dynamics computed at the CoM of the system [41]. A momenta task can be formulated by using the centroidal momentum matrix $A_G(\mathbf{q})$ which is the product of the inertia matrix of the system and the Jacobian matrix of the system ($A_G(\mathbf{q}) = I_{\text{sys}} J_{\text{sys}}$) [40], which maps the joint velocities $\dot{\mathbf{q}}$ to the the centroidal momenta \mathbf{h}_G as follows:

$$\mathbf{h}_G = A_G(\mathbf{q}) \dot{\mathbf{q}}. \quad (11)$$

This relation has the form of Eq. (8). Moreover Eq. (11) can be differentiated with respect to time as:

$$\dot{\mathbf{h}}_G = A_G(\mathbf{q}) \ddot{\mathbf{q}} + \dot{A}_G(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}, \quad (12)$$

which matches the pattern of Eq. (9).

A task can be expressed directly in the configuration space to control the posture of the robot. Instead of the task Jacobian matrix, a selection matrix is used to select the joints that will be controlled.

3.3 Inverse Kinematics Control

The inverse kinematics problem consists in finding the joint kinematics that allows the robot to accomplish a reference kinematic task behavior. In the next subsections we show how to control the execution of tasks expressed in terms of velocities (first order kinematics) and accelerations (second order kinematics).

3.3.1 First order kinematics

Solving the first order kinematics comes to determine suitable joint velocities to generate the desired task velocity $\mathbf{e}(\mathbf{q}, \dot{\mathbf{q}})$ using the relationship obtained in Eq. (8). In order to control the task performance, a reference task behavior \mathbf{e}^* is provided as input and the control problem comes to solve the following unconstrained minimization problem:

$$\min_{\dot{\mathbf{q}}^*} \|\dot{\mathbf{e}}^* - J_e(\mathbf{q})\dot{\mathbf{q}}^*\|_2^2. \quad (13)$$

The solution to this problem provides the control law of the system as follows:

$$\dot{\mathbf{q}}^* = J_e^\#(\mathbf{q})\dot{\mathbf{e}}^* + P_{J_e}\dot{\mathbf{q}}_2, \quad (14)$$

where $\{\cdot\}^\#$ represents the generalized inverse, P_{J_e} is the projector onto the null space of $J_e(\mathbf{q})$ (e.g. $P_{J_e}J_e = 0$ and $P_{J_e}P_{J_e} = P_{J_e}$) and $\dot{\mathbf{q}}_2$ is a secondary control input that can be used to exploit the systems redundancy with respect to the task.

3.3.2 Second order kinematics

The second order kinematics problem deals with the relationship between task acceleration and joints acceleration provided by Eq.(9). The control problem, which consists in finding the suitable joint accelerations that generates the task reference behavior, can be expressed by the following minimization problem:

$$\min_{\ddot{\mathbf{q}}^*} \|\ddot{\mathbf{e}}^* - J_e(\mathbf{q})\ddot{\mathbf{q}}^* + \dot{J}_e(\mathbf{q})\dot{\mathbf{q}}\|_2^2. \quad (15)$$

Using the same notation as before, the control law expressed in terms of the joint accelerations is then written as:

$$\ddot{\mathbf{q}}^* = J_e^\#(\mathbf{q})(\ddot{\mathbf{e}}^* + \dot{J}_e(\mathbf{q})\dot{\mathbf{q}}) + P_{J_e}\ddot{\mathbf{q}}_2. \quad (16)$$

3.4 Inverse Dynamics Control

The inverse dynamics problem aims at determining the suitable joint torques to generate a reference task acceleration behavior $\ddot{\mathbf{e}}^*$. By multiplying the actuated part of Eq. (7) by JM^{-1} and replacing Eq. (9) in Eq. (7) we obtain the following relation:

$$\ddot{\mathbf{e}} + J_e(\mathbf{q})M^{-1}\mathbf{b} - \dot{J}_e(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = J_e(\mathbf{q})M^{-1}\boldsymbol{\tau}. \quad (17)$$

The inverse dynamics control law can be written as:

$$\boldsymbol{\tau}^* = (J_e(\mathbf{q})M^{-1})^\#(\ddot{\mathbf{e}}^* + J_e(\mathbf{q})M^{-1}\mathbf{b} - \dot{J}_e(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}) + P_{J_eM^{-1}}\boldsymbol{\tau}_2, \quad (18)$$

where $(P_{J_eM^{-1}})$ is the projector onto the null space of $J_e(\mathbf{q})M^{-1}$ and $\boldsymbol{\tau}_2$ is an arbitrary vector that can be used to control other tasks. Eq. (18) can be extended to include rigid contact constraints [43]. The control of humanoid robots interacting with the environment has to take into account external forces. Thus, the control problem consists in finding a control law that achieves a desired task behavior while respecting the dynamic model of the system and additional constraints that ensure the feasibility of the motion. This problem can be solved by setting a minimization problem

under equality and inequality constraints as in [42]:

$$\begin{aligned}
& \min_{\dot{\mathbf{q}}, \boldsymbol{\tau}, \boldsymbol{\lambda}} \|\ddot{\mathbf{e}}^* - \ddot{\mathbf{e}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\|_N^2 \\
s.t. \quad & M(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{g}(\mathbf{q}) - \sum_{k=1}^K J_k^T(\mathbf{q})\boldsymbol{\lambda}_k = S^T \boldsymbol{\tau}_{int} \\
& J_k \ddot{\mathbf{q}} + \dot{J}_k \dot{\mathbf{q}} = 0 \\
& \boldsymbol{\lambda}_k^\perp \geq 0
\end{aligned} \tag{19}$$

where J_k is the contact Jacobian associated to the k^{th} contact point. The first equality of Eq. (19) ensures the respect of the dynamic model of the system. The inequality constraint $\boldsymbol{\lambda}_k^\perp \geq 0$ guarantees that the contact forces are correctly oriented and that there is no interpenetration (rigid contact). In the same manner, other inequality constrains can be added such as joint limits ($\underline{\mathbf{q}} \geq \mathbf{q} \geq \bar{\mathbf{q}}$), torque limits ($\underline{\boldsymbol{\tau}} \geq \boldsymbol{\tau} \geq \bar{\boldsymbol{\tau}}$) or other tasks (as given in subsection 3.2).

3.4.1 Hierarchical Control

Solutions to problems in the form of Eq. (19) can be formulated based on null space projections or optimization methods. Prioritization schemes are based on projections onto the null space of higher order priority tasks in the form of Eq. (14) and Eq. (16). Unfortunately, these methods do not allow to cope with inequality constraints. Instead, numerical optimization techniques can be used to solve problems of the form of Eq. (19). A typical method is to use Hierarchical Quadratic Programming (HQP) [29, 17]. In order to compute HQP, basis multiplication was applied according to [16]. Computational details about this approach are given in the sequel:

Basis Multiplication

Linear equality constraints of the form $\mathbf{A}\mathbf{x} = \mathbf{b}$ such as task functions, can be solved using hierarchical control. Let us consider the case of L linear constraints (e.g. tasks) $(A_1, \mathbf{b}_1) \cdots (A_l, \mathbf{b}_l) \cdots (A_L, \mathbf{b}_L)$ that have to be satisfied at best and let us consider that constraints are conflicting between them. A strict hierarchy of constraints can be used to solve this problem [47]. The constrain with the highest priority (A_1, \mathbf{b}_1) can be solved at best in a least-square sense with the pseudo-inverse. Then the second constraint (A_2, \mathbf{b}_2) is solved in the null space of the first constraint. The generic solution to solve the p levels of the hierarchy can be written as:

$$\mathbf{x}_l^* = \sum_{l=1}^L (A_l P_{l-1})^+ (\mathbf{b}_l - A_l \mathbf{x}_{l-1}^*) + \tilde{P}_L \tilde{\mathbf{x}}_{L+1}, \tag{20}$$

with $P_0 = \mathbb{1}$, $x_0 = 0$ and $\tilde{P}_L = P_{L-1}P_L$ is the projector in the null space of $(A_l P_{l-1})$, x^* denotes the solution for the hierarchy of constraints composed of L linear constraints, P is a projector onto the null space of A ($AP = 0$ and $PP = \mathbb{1}$) and \tilde{x}_{L+1} is any vector of the configuration space that can be used to accomplish another objective. In order to fasten the numerical resolution of Eq. 20 a basis multiplication approach has been proposed [16, 17]. Given a basis Z_1 of the null space of A_1 ($A_1 Z_1 = 0$), the projector in the null space of A_1 can be written as $P_1 = Z_1 Z_1^T$. Eq. 20 can be rewritten as:

$$\mathbf{x}_l^* = \sum_{l=1}^L Z_{l-1} (A_l Z_{l-1})^+ (\mathbf{b}_l - A_l \mathbf{x}_{l-1}^*) + \tilde{Z}_L \tilde{\mathbf{x}}_{L+1}, \quad (21)$$

which is more efficient to compute than Eq. (20) due to the size of the matrices. Note that Eqs. (14), (16) and (18) can be mapped to Eqs. (20) and (21).

4 Case study

Future generations of robots will include agile, robust, efficient and powerful robots that perform more dynamic tasks. The utilization of such kind of robots will require understanding how dynamic motions have to be generated. One of the choices is, without doubt, to take inspiration from humans. As previously mentioned, the framework described in this chapter can be applied to a wide range of motions without loss of generality. This section presents a practical case of application to a highly dynamic and complex Parkour motion — called Parkour precision technique. First Parkour is introduced and the state of the art related to the Parkour precision technique is presented. Then, a skeletal model that is used for analyzing and generating the motion is described. Afterwards, the biomechanics methodology used for analyzing the Parkour precision technique is given. Finally, this analysis is used to parameterize a hierarchical controller in order to simulate similar Parkour motion with the skeletal model.

4.1 Biomechanics of Parkour landing

Parkour is a discipline where movements are highly dynamic and complex. It requires practitioners — called traceurs — to adapt their motion to the environment in order to overcome obstacles quickly and efficiently. Motion strategies are derived from a military method developed in France after World War I, which was inspired by natural movements observed in skilled indigenous African tribes [25]. This method includes a combination of motions with variations of jumping, landing, climbing and vaulting strategies. Common to most Parkour techniques are the jumping and landing strategies. Mastering jumping and landing techniques allows

practitioners to execute efficient and safe motions. In our study, we analyzed the Parkour precision jumping and landing strategies (Fig. 2).

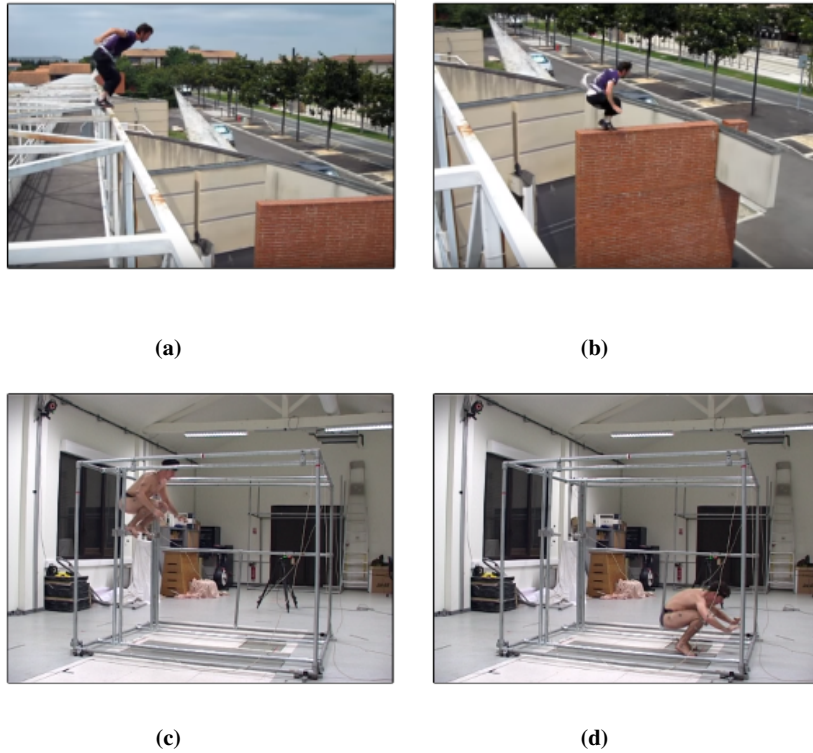


Fig. 2: Parkour precision jumping and landing techniques performed in a urban space in (a) and (b), and inside a motion capture laboratory in (c) and (d).

4.1.1 Parkour precision jump and landing techniques

Precision landing is commonly used for relatively low dropping heights. For executing this motion, trainers instruct practitioners to land and stay with precision on their forefoot avoiding heel contact with the ground, bend their lower limbs without any varus-valgus motion of the knees, and use their arms to counterbalance the movement and stabilize themselves. In what follows, we explain the selection of the performance criteria used to study the Parkour motion, which allowed us to identify the tasks for generating the movement using the robotic framework of hierarchical task control.

The considered Parkour motion can be decomposed into three phases: take-off, flight and landing (in subsection 4.1.5 we will detail how motion phases were divided). The take-off phase of the Parkour technique is similar to the standing long jump technique used in athletics. It requires practitioners to perform an horizontal jump without a preparatory running. Previous studies on long jump technique have suggested that during take-off motion, the principal performance factor is the velocity of the center of mass "CoM" [51] which defines the ballistic trajectory during the flight phase. Another performance criterion that has been proposed, is the angular momentum generated with the upper limbs. It was shown that using arms to generate angular momentum around the medial-lateral axis contributes to reduce the torques requested to the lower-limb joints [6]. This means that a person can counteract gravity effects by using the angular momentum generated with the upper limbs, whereas lower limbs will be more involved in the production of the force to project the CoM. This strategy contributes also to avoid excessive forward rotation [3] positioning the body in preparation to the landing phase [2]. According to the momenta conservation principle, the angular momentum remains constant during the flight phase. As a consequence the taking-off controller that generates the flight motion must be parameterized consistently. On the other hand, during the flight phase the linear momentum is only affected by the gravity. The landing phase of the Parkour technique induces practitioners to decrease peak ground reaction forces "GRFs" [48] and to stabilize better [36]. Decreasing vertical GRF and controlling stability through the antero-posterior and medial-lateral components of the GRFs is equivalent to control the derivative of linear momentum. Note that the derivative of momenta is equal to the net external forces/torques according to Euler's laws of motion. To avoid falling down after a highly dynamic landing, angular momentum might also contribute to stability by reducing the angular accelerations at the CoM. In fact, during our pre-tests, we observed that practitioners swung using their upper limbs in a three dimensional fashion.

4.1.2 Whole-body model

The choice of the physical model depends on the type of analysis. We are interested in a whole-body 3D analysis able to reproduce highly dynamic Parkour movements. To this end, we decided to use a whole-body 3D model that represents the main elementary movements of a Parkour jumper including 42 DoF (Fig. 1b). Note that a 3D model is necessary, because as shown by previous studies [26], a sagittal model is not sufficient to study the upper body motion during standing long jumps. The characteristics of the model are presented below:

- The lower limb, pelvis and upper limb anthropometry are based on the running model of Hamner et al. [22]. Mass properties of the torso and head segments (the head and neck segments were modeled as one segment) are estimated from the regression equations of Dumas et al. [10, 11]. Hands anthropomorphic data are based on regression equations of De Leva [34].

- Each lower extremity has seven DoF. The hip is modeled as a ball-and-socket joint, the knee is modeled as a revolute joint, the ankle is modeled as 2 revolute joints (flexion-extension and inversion-eversion) and the toes with one revolute joint at the metatarsals.
- The pelvis joint is modeled as a free flyer joint to permit the model to translate and rotate in the 3D space. This 6D joint is attached to the free-floating base (root frame) of the under-actuated systems as described in Section 3.1. The lumbar motion is modeled as a ball-and-socket joint [1] and the neck joint is also modeled as a ball-and-socket joint.
- Each arm includes 8 DoF. The shoulder is modeled as a ball-and-socket joint, the elbow and forearm rotations are modeled with revolute joints to represent flexion-extension and pronation-supination [28], the wrist flexion-extension and radial-ulnar deviations are modeled with revolute joints, and the hand fingers are modeled with one revolute joint for all fingers.
- The model includes a whole-body marker set with 48 markers placed in anatomical landmarks as suggested by Wu et al. [53, 54]. The visual elements are also based on the running model of Hamner et al. [22].

4.1.3 Experimental Protocol

Five healthy trained male traceurs (age: 22.2 ± 4.8 y, height: 1.73 ± 0.04 m, mass: 66.6 ± 5.1 kg) volunteered for the study. The traceurs' experience in Parkour practice was 5.4 ± 2.1 years. The subject exclusion criterion was based on history of lower extremity injuries or diseases that might affect jump and landing biomechanics. The experiments were conducted in accordance with the standards of the Declaration of Helsinki (rev. 2013) and approved by a local ethics committee. Participants performed a warming up session followed by a familiarization period during which the protocol instructions were provided to them, and during which the participant familiarized with the lab environment. The landing protocol was designed to include a jump height of 75 % of the height of the participant and a landmark placed at a horizontal distance equal to the square of the jump height (See Eq. (22)). Participants were asked to land on the target specified by the landmark and to stabilize using the Parkour precision technique.

4.1.4 Data Acquisition

A total of 8 successful repetitions per participant were recorded. Whole-body 3D kinematic data were collected using 14 infra-red cameras sampling at 400 Hz (Vicon, Oxford Metrics, Oxford, UK) and recording 48 reflective markers placed on the participant's body. Two force plates (AMTI, Watertown, MA, USA) embedded into the floor in order to record landing GRFs and two rigid handle bar sensors (SENSIX, Poitiers, Vienne, France) with a diameter of 63 mm placed on a Parkour tubular structure to record take-off GRFs, were used sampling at 2000 Hz. Force

data were used to define the onsets used to divide the Parkour motion into phases. Markers were located on the participants body based on Wu and Dumas recommendations [10, 11, 53, 54] as follows: the first and fifth metatarsal, second toe tip, calcaneus, lateral and internal malleolus, anterior tibial tuberosity, lateral and medial epicondyles of knee, greater trochanter, posterior superior iliac spine and anterior superior iliac spine, procesuss xiphoideus, incisura jugularis, seventh cervicale, tenth thoracic vertebra, acromioclaviculare, medial and lateral epicondyle, ulnar and radial styloid, second and fifth metacarpal heads, second fingertip, sellion, occiput, right and left temporal (Fig. 1).

4.1.5 Data Analysis

Kinematics and kinetics were processed with the same cut-off frequency [31] using a low-pass Butterworth digital filter of 4th order applied in a zero-phase. A cut-off frequency of 35 Hz was used after a residual analysis [52]. All computations were performed using a custom made program with a whole-body model and a physics engine. Inverse kinematics computations were solved by minimizing the squared distance between recorded and virtual markers (Eq. (4)) and using Euler xyz body-fixed rotation angles [46] with body frames defined according to ISB recommendations [53, 54].

The motion was divided into three phases: take-off, flight, and landing (Fig. 2). The take-off phase was defined from the minimum vertical position of the CoM until the last foot contact. The flight phase was defined between the end of the take-off phase until the initial contact "IC" with the ground, that we identified as the instant when the vertical ground reaction force reached 50 N. The landing phase was defined from IC until the CoM reached its minimum vertical position. Each phase was normalized by its time duration from 0% to 100% (Fig. 2). The linear momentum was normalized by the participant weight, and the angular momentum was normalized by the participant weight and height. Mean and standard deviation were calculated for each participant and for the whole group. Group means at key frames were used to parameterize the algorithm for motion generation.

4.2 Motion Generation

In order to generate the motion, we utilized the skeletal model introduced in section 4.1.2 for human motion analysis. The robotics framework of hierarchical task control introduced in section 3 was considered. Tasks were set and parameterized based on the biomechanical study made in section 4.1. The motion was generated by using the method of null space projections and second order kinematics introduced in section 3. The rigid body dynamics computations were done by using the Pinocchio library developed by the Gepetto team of LAAS-CNRS [4].

4.2.1 Description of the Tasks

As for motion analysis, the motion generation is divided into three phases: take-off, flight and landing. In order to apply the hierarchical control framework, tasks are stacked in a hierarchical manner for each motion phase. The selection of tasks and their hierarchy are based on the previous biomechanical study using mainly linear and angular momentum tasks. Additional tasks for foot placement, CoM and posture at key points (beginning/end of each motion phase) were also considered to better specify the motion. The desired values for parameterizing the controller were deduced from the biomechanical study. The motion was controlled using joint accelerations according to Eq. (12).

Preparation phase

A preparation phase was added before to the take-off phase for parameterizing the simulation with the initial conditions. Note that this phase is not strictly necessary as the controller can be parameterized directly with the initial conditions of the take-off phase which can be obtained from the motion analysis. This motion is generated through the control of a hierarchy of tasks organized as follows:

- At the highest priority we set a 3D foot placement task. This task was used to control the 3D position of each forefoot so that contact with the handle-bars can be maintained during this phase. The orientation components of these tasks were not constrained.
- At the second level of the hierarchy, we set a task specifying a desired 3D position of the CoM which was deduced from motion analysis.
- The remaining DoF were used to control the whole-body posture which was also extracted from motion analysis.

Take-off phase

The take-off phase follows the preparation phase. The motion generated during this phase is organized in terms of tasks as follows:

- At the highest priority level, the 3D foot placement task of the preparation phase is kept.
- To generate the ballistic trajectory of the CoM, a linear momentum task is added at the second level of the hierarchy. The antero-posterior and vertical components provide the modulation of the CoM velocity for generating a desired ballistic trajectory. The medial-lateral component is regulated to zero to avoid undesired deviations of the CoM trajectory during the flight phase.
- In order to control the angular momentum at the CoM, a third task is stacked. This task imposes zero momentum around the vertical and antero-posterior (A-P) axes, and a desired angular momentum around the medial-lateral (M-L) axis.

It allows the body model to reach a desired posture before landing and avoid somersaults. This task should also alleviate torques at the lower limbs as suggested in the literature (see subsection 4.1).

Flight Phase

When the velocity and take-off angle of the CoM trajectory are suitable to reach a desired horizontal distance, the top level task of the foot placement is removed from the stack of tasks and the flight phase begins. The horizontal distance d_{flight} that the CoM will travel is calculated during the take-off phase according to the ballistic equations as:

$$d_{flight} = \frac{v \cos \theta}{g} \left(v \sin \theta + \sqrt{(v \sin \theta)^2 + 2gh_0} \right), \quad (22)$$

where v is the initial speed of the CoM (before the flight phase begins), θ is the take-off angle, g is the gravity acceleration and h_0 is the initial height of the CoM. During the flight phase, the momentum is conserved. A second level task is added to impose a desired posture before contacting the ground.

Landing Phase

The landing phase starts at the end of the flight phase and is set as follows:

- At the highest level of the hierarchy, the vertical feet position is regulated to keep the current contact position with the ground. A desired flexion of the toes and ankle joints is also imposed.
- At the second level, a 3D linear momentum task is added to decrease the velocity of the CoM to zero.
- At the third level, a task is added to regulate angular momentum to zero in order to avoid tipping motions of the model.
- At the fourth level of the hierarchy, a task is added to keep the CoM inside the vertical projection of the support polygon (medial-lateral and antero-posterior axis) in order to provide a static equilibrium state until the end of the motion.

4.3 Temporal Sequence

The chronological sequence of tasks stacked during each motion phase with their hierarchical order is depicted in the following table:

Hierarchy	Preparation	Take-off	Flight	Landing
1	Feet		Momenta	Feet
2	CoM	Linear Mom.	Posture	Linear Mom.
3	Posture	Angular Mom.		Angular Mom.
4				CoM

Table 1: Hierarchy of tasks used for generating motion in each phase.

4.4 Tasks behavior

Exponential decays of the task function were used to specify the task behavior. Weighting matrices multiplying the control input (\ddot{q}^*) were tuned for the take-off and landing phases in order to apply higher gains to lower joints when controlling the linear momentum, and higher gains to upper joints when controlling the angular momentum in accordance to our biomechanical study. As the trunk segment has a significant mass compared to the upper limbs, the gain for the lumbar flexion was lowered to avoid undesirable behaviors of momenta.

4.5 Results

The experimental results and the generated motions present strong similarities, specially when comparing the motion of the lower body. Fig. 3 shows snapshots of the human motion reconstructed by means of the inverse kinematics method of biomechanics, and snapshots of the motion generated using the hierarchical task control. The profiles of the linear and angular momenta of the experimental and generated motions are shown in Fig. 4 and in Fig. 5. The momenta profiles of humans represent the mean behavior of the Parkour experts with the corresponding standard deviation. Table 2 shows the difference between the ranges of motion (RoM) of the human group and the simulation model for the principal segments in the motion during each phase. In the sequel we refer to "the humans" to describe the average human motion and "the model" to refer to the generated one.



Fig. 3: Snapshots of the take-off phase (a)(b)(c), flight phase (d), and landing phase (e)(f)(g)(h) at different percentages of the motion phases. The skeleton on the left, represents the result of the inverse kinematics from motion analysis of a Parkour practitioner, whereas the skeleton on the right is the motion generated through hierarchical control.

	Take-off		Flight		Landing	
	RoM [deg]		RoM [deg]		RoM [deg]	
	L	R	L	R	L	R
Neck-head flexion-extension		19		22		15
Trunk flexion-extension		-20		-10		-5
Upper arm flexion-extension	-1	-1	65	66	-44	-30
Upper arm abduction-adduction	17	33	-2	5	10	9
Upper arm rotation	17	24	44	68	47	54
Forearm flexion-extension	-17	-24	-9	-4	-15	3
Thigh flexion-extension	-2	-2	16	22	2	0
Thigh adduction-abduction	-5	-9.6	1	-4	2	2
Thigh rotation	-6	0	1	9	8	6
Shank flexion	12	12	30	44	20	23

Table 2: Difference in the ranges of motion (RoM) in degrees, of the analyzed human motions and the generated motion. The table shows the most relevant coordinates during the take-off, flight and landing phases of the Parkour technique. Negative values mean that the RoM of the generated motion is higher than the RoM of the human experts.

4.5.1 Take-off

- At the beginning of the motion, linear momentum values were different between the humans and the model. Throughout the motion, the linear momentum was similar in the medial-lateral component while the antero-posterior and vertical component of the linear momentum behaved differently. More antero-posterior linear momentum was generated by the humans at the end of the take-off phase, and more vertical linear momentum was generated by the model at the end of the motion phase.
- Although at the beginning of the motion angular momentum values were different, angular momentum behaved similarly in the humans and the model in the sequel. The angular momentum components around the antero-posterior and vertical axes were almost zero at the end of the motion phase, while the angular momentum component around the medial-lateral axis was not zero at the end of the motion phase.
- Fig. 3 shows that upper-limbs were coordinated differently with the time evolution and that the trunk and hips were more flexed in humans.
- RoM of all coordinates were similar (Table 2). RoM of thigh abduction-adduction and forearm flexion-extension were slightly higher with the model, while upper arm abduction-adduction and rotation appear to be higher with the humans.

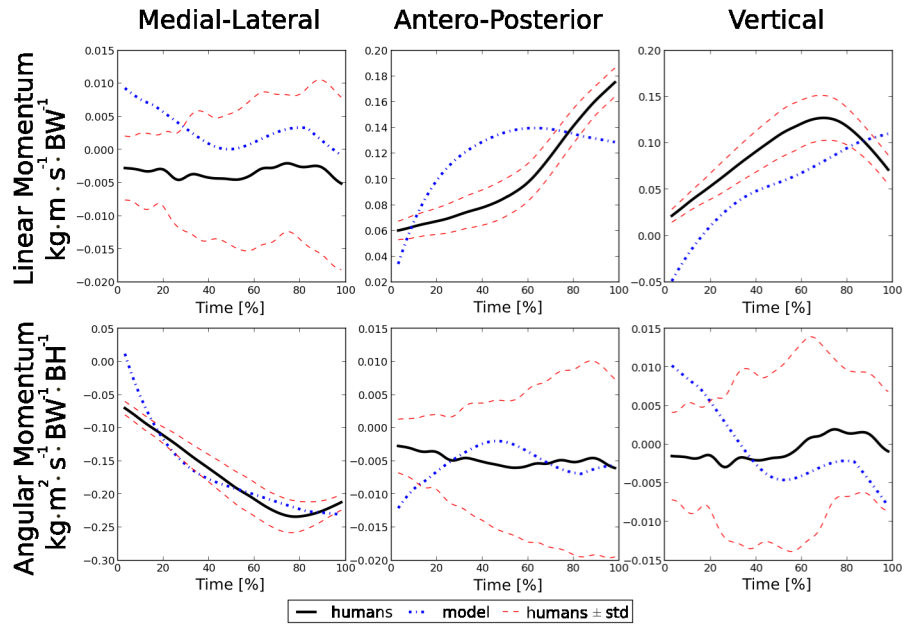


Fig. 4: Momenta profiles of the humans (\pm SD), and of the simulation model during the take-off phase. The first row shows the medial-lateral, antero-posterior and vertical components of the linear momentum normalized by the body weight. The second row shows the angular momentum normalized by the body weight and height, about the the medial-lateral, antero-posterior and vertical axis.

4.5.2 Flight

Momenta were not compared during this motion (see subsection 4.1.1). Figs. 3c and 3d show that the motion looks different at the beginning and at 50 % of the flight. RoM were also different, specially in the case of the upper limb movement and the shank flexion-extension (Table 2).

4.5.3 Landing

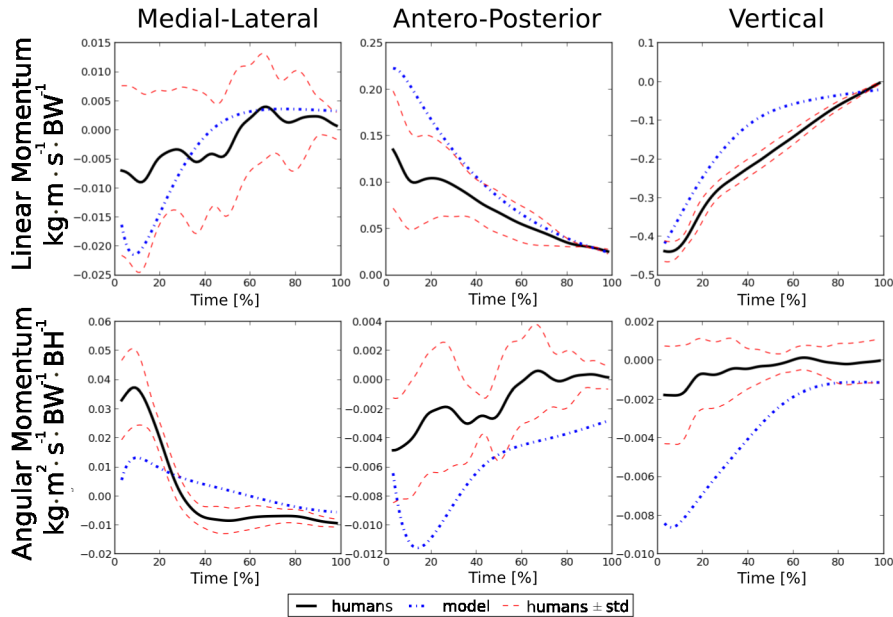


Fig. 5: Momenta profiles of the humans (\pm SD), and of the simulation model during the landing phase. In the first row: medial-lateral, antero-posterior and vertical components of the linear momentum normalized by the body weight are displayed. In the second row: angular momentum normalized by the body weight and height around the the medial-lateral, antero-posterior and vertical axis are displayed.

- Linear momentum profiles of the humans and the model were globally similar.
- At the beginning of the landing phase, the model generated less angular momentum around the medial-lateral axis, while higher angular momentum was observed around the antero-posterior and vertical axis.
- From Fig. 3, we can observe that the humans and the model landed with a similar posture. At 25 % and 50 % of the landing phase, the trunk of the model is more flexed than the the trunk of humans.
- RoM of the upper arm coordinates appear to differ slightly while the RoM of the other coordinates look similar.

4.6 Discussion

The analyzed and the generated motion were compared in terms of kinematics and momenta. The results showed that the kinematics of the humans group were similar to the kinematics of the simulated model, specially for the lower limbs. Time evolution of momenta was sometimes slightly different during the take-off phase, while it was comparable during the landing phase. Whole-body coordination was congruent between the humans and the model, although the upper limbs strategy did not evolve similarly with time. In the next subsections we analyze the results per phase.

4.6.1 Take-off

During the take-off phase, we observed more antero-posterior linear momentum in the motions generated with the model. For generating the desired ballistic profile, the model compensated for the lack of antero-posterior linear momentum by increasing the vertical linear momentum. This strategy allowed the model to reach the initial CoM speed for landing at the requested distance. This increase of angular momentum of the model with respect to the humans one might explain why the model jumped with the trunk more extended than the human (Fig. 3c), and why the RoM of the trunk was higher. The time behavior of the vertical and antero-posterior linear momentum was different in the model. This might be due to the task reference behavior which imposes an exponential decay of the error between the actual and desired value (as explained in section 3).

The profiles of angular momentum were similar for the humans and the model. The arms, which contribute to angular momentum, behaved differently with time evolution as shown in Fig. 3. The reason might be that the upper limbs contributed also to increase the vertical linear momentum (which behaved differently in the model). Note also that the RoM of the forearm flexion-extension motion was higher in the simulation model. In spite of this, the model jumped with an angular momentum that allowed it to prepare properly the body posture for landing (Fig. 3e). Controlling the posture before landing allows for a better control of stability and impact damping (See subsection 4.1).

On angular momentum control during the take-off phase

We carried on simulations by decreasing and increasing the desired angular momentum during the take-off phase. The results showed that when the angular momentum is decreased, the model lands with a posture that makes it fall forwards (Fig. 6a). Conversely, if the angular momentum is increased, the model lands with a posture that makes it fall backwards (Fig. 6b). These results highlight the importance of controlling the angular momentum when jumping.

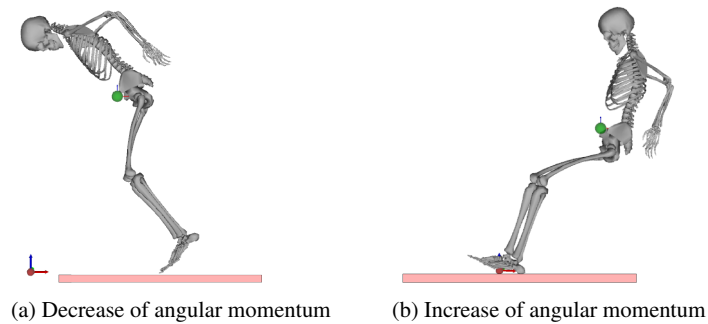


Fig. 6: Effects on posture at landing when modifying the desired angular momentum during the take-off phase. The motion was generated by using the same hierarchical controller. Only the desired values of the angular momentum were modified during the take-off phase.

4.6.2 Flight

The kinematics of the humans and the model were different during the flight phase. This might be due to the fact that the flight motion of the model was generated without constraining the trunk which has the highest mass. Thus, segments with small masses might have contributed less importantly to keep momenta constant during flight. In fact, the RoM of the trunk was higher in the model whereas the RoM of the upper arm coordinates was higher in the humans. Note that the final posture (before landing) does not reflect the higher excursion of the upper arm coordinates in the humans. Instants before landing, arms are swung backwards. Later on, when contacting the ground, arms have already been swung forwards in preparation to the landing phase. In the model, this swing strategy of the arms was not considered.

4.6.3 Landing

Linear and angular momentum were similar in the humans and the model. Nevertheless, upper-body coordination during the landing phase looks different (Fig. 3). The model landed with the trunk more flexed than humans. Thus, upper-limbs might compensate for this strategy by generating a counterbalancing angular momentum. In fact, the RoM of the upper arm flexion-extension was higher in the model. It turns out that the considered decay rate and hierarchy of tasks allowed the model to replicate the evolution of the momenta in humans.

5 Discussion

In this chapter, we presented an interdisciplinary methodology for generating human inspired motions with anthropomorphic systems such as robots. The approach involves robust biomechanics methods to analyze the human movements and the robotics framework to generate the motion. We showed that our approach is suitable to reproduce highly dynamic and complex motions comparable to those of humans. In this final section we summarize our methodology, we provide a discussion about the interest on using the proposed methodology, we show the limitations of the proposed approach, we give some perspectives for the future work, and we conclude with remarks about the interpretation of our results.

5.1 Summary of the method "from Biomechanics to Robotics"

Important aspects of this approach can be summarized as follows:

- A physical model suitable for analyzing and/or generating the motion of interest is selected/created.
- An experimental protocol for motion analysis is designed and the human movements are recorded by using motion-capture techniques.
- The recorded motion is analyzed using robust biomechanics techniques and key performance variables of the movement are identified. Performance variables provide the information needed to create a set of tasks.
- By understanding how the identified performance variables favors the motion generation in humans, the set of tasks is organized in a hierarchical manner according to their importance in the execution of the motion. A stack of tasks is created.
- The robot controller is further parameterized using information from control strategies observed in humans by weighting the controller input. This information can be obtained from biomechanical studies.
- The motion is generated through the task function approach in robotics using the identified tasks and the physical model for motion generation.
- The human motion and the motion artificially generated are compared using the same physical model created for analyzing the motion.

5.2 Interest of the proposed approach

There are interesting aspects of this approach that deserve further discussion. First, our methodology differs from conventional task-space approaches, because it allows for generating whole-body motion based on the biomechanics of human movement. Quantifying and understanding the mechanics of human motion offers the possibil-

ity to parameterize robotic algorithms based on human expertise to execute movements. This provides a way to generate more robust and efficient motions. Furthermore, generating anthropomorphic motions that look more human-like is of great interest for human-robot interaction. Second, comparing the analyzed human motion with the motion generated based on the robotics task function approach provides an interesting tool for validating hypotheses about the organization of human movement based on task hierarchies.

5.3 Limitations and future work

Though the biomechanics methodology provides an efficient tool for analyzing human movement, it is known to have some limitations. Some source of errors are the simplifications made when modeling the human body (e.g. modeling segments as rigid bodies, number of DoF of the model, or simplifications for modeling complex human articulations), the estimation of anthropomorphic parameters and the data processing (e.g. motion reconstruction, filtering of the noise, soft tissue artifacts). The feasibility of the motion generated can be assessed in terms of joint torque and joint power by adding inequality constraints (Eq. (19)). To this end, the hierarchical controller can be designed using quadratic programming QP as suggested in subsection 3.4.1. A deeper analysis between the human and the model strategies to compare other performance variables such as reactions forces, torques, power and energy dissipation is also desirable. Furthermore, the behavior of the tasks (decay rate of the task function) can also be modified to generate smoother trajectories (see for example [27]). A vision task [5] could also be added to reflect the importance of vision in humans when performing this type of motions. Other approaches can be used to generate highly dynamic and complex motions by considering the whole trajectory along a finite time horizon, e.g. optimal control. In that case hybrid cost functions, usually described as weighted-sums of elementary criteria, are considered. The difficulty is then to identify the set of weights that lead to the best replication of the observed human movement.

Finally, though our results suggest that the proposed approach is suitable to generate motion similar to that of humans, we are not attempting to propose that humans compute inverse kinematics or inverse dynamics problems. This is still an open question, which is outside the scope of this chapter. Nevertheless, we could point out that strategies of task control have been observed in human motion. It was shown that the central nervous system finds stable solutions of motor tasks in accordance with the uncontrolled manifold theory "UCM" [45] and the motor abundance principle [20]. It is also suggested that hierarchies in human motion appear at different levels such as neurons, muscles, joints and tasks [33, 20]. Lastly, the UCM theory is linked to the equilibrium point [18] which is a physiological approach to understand human motions.

Acknowledgements Part of this work is supported by ANR Entracte project (ANR 13CORD-002-346 01), the European Research Council for the project Actanthrope (ERC-ADG347 340050) and the Flag-Era European project RoboCom++.

References

1. Frank C Anderson and Marcus G Pandy. A Dynamic Optimization Solution for Vertical Jumping in Three Dimensions. *Computer Methods in Biomechanics and Biomedical Engineering*, 2(3):201–231, 1999.
2. Blake M. Ashby and Scott L. Delp. Optimal control simulations reveal mechanisms by which arm movement improves standing long jump performance. *Journal of Biomechanics*, 39(9):1726–1734, 2006.
3. Blake M. Ashby and Jean H. Heegaard. Role of arm motion in the standing long jump. *Journal of Biomechanics*, 35(12):1631–1637, 2002.
4. Justin Carpentier, Florian Valenza, Nicolas Mansard, and Others. Pinocchio: fast forward and inverse dynamics for poly-articulated systems. <https://stack-of-tasks.github.io/pinocchio>, 2015.
5. François Chaumette and S. Hutchinson. Visual servo control, Part I: Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4):82–90, 2006.
6. Kuangyou B. Cheng, Chih Hung Wang, Hui Chuan Chen, Chin Dai Wu, and Hung Ta Chiu. The mechanisms that enable arm motion to enhance vertical jump performance—A simulation study. *Journal of Biomechanics*, 41(9):1847–1854, 2008.
7. L. Chèze, B.J. Fregly, and J. Dimnet. A solidification procedure to facilitate kinematic analyses based on video system data. *Journal of Biomechanics*, 28(7):879 – 884, 1995.
8. 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. *IEEE Transactions on Biomedical Engineering*, 54(11):1940–1950, 2007.
9. R. Dumas. Influence of the 3D Inverse Dynamic Method on the Joint Forces and Moments During Gait. *Journal of Biomechanical Engineering*, 129(5):786, 2007.
10. R. Dumas, L. Chèze, and J. P. Verriest. Adjustments to mcconville et al. and young et al. body segment inertial parameters. *Journal of Biomechanics*, 40(3):543–553, 2007.
11. R. Dumas, L. Chèze, and J.-P. Verriest. Corrigendum to Adjustments to McConville et al. and Young et al. body segment inertial parameters [J. Biomech. 40 (2007) 543553]. *Journal of Biomechanics*, 40(7):1651–1652, 2007.
12. Sonia Duprey, Laurence Chèze, and Raphaël Dumas. Influence of joint constraints on lower limb kinematics estimation from skin markers using global optimization. *Journal of Biomechanics*, 43(14):2858–2862, 2010.
13. JL Durkin, JJ Dowlingm, and DM Andrews. The measurement of body segment inertial parameters using dual energy X-ray absorptiometry. *Journal of Biomechanics*, 35(12):1575–1580, 2002.
14. Rainald M Ehrig, William R Taylor, Georg N Duda, and Markus O Heller. A survey of formal methods for determining the centre of rotation of ball joints. *Journal of Biomechanics*, 39(15):2798–809, 2005.
15. Rainald M Ehrig, William R Taylor, Georg N Duda, and Markus O Heller. A survey of formal methods for determining functional joint axes. *Journal of Biomechanics*, 40(10):2150–7, 2007.
16. A. Escande, N. Mansard, and P. B. Wieber. Fast resolution of hierarchized inverse kinematics with inequality constraints. In *2010 IEEE International Conference on Robotics and Automation*, pages 3733–3738, May 2010.
17. Adrien Escande, Nicolas Mansard, and Pierre-Brice Wieber. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 33(7):1006–1028, 2014.

18. Anatol G. Feldman. Functional tuning of the nervous system with control of movement or maintenance of a steady posture. II. Controllable parameters of the muscle. *Biophysics*, 11:565–578, 1966.
19. Tamar Flash. The Coordination of Arm Movements: Mathematical Model'. *The Journal of Neuroscience*, 5(7):1688–1703, 1985.
20. G Gera, S Freitas, M Latash, K Monahan, G Schoner, and J Scholz. Motor Abundance Contributes to Resolving Multiple Kinematic Task Constraints. *Motor Control*, 14(1):83–115, 2010.
21. Sovannara Hak, Nicolas Mansard, Oscar Efrain Ramos Ponce, Layale Saab, and Olivier Stasse. Capture, Recognition and Imitation of Anthropomorphic Motion. In *IEEE International Conference on Robotics and Automation (ICRA 2012)*, pages pp. 3539–3540, St Paul, United States, May 2012.
22. Samuel R. Hamner, Ajay Seth, and Scott L. Delp. Muscle contributions to propulsion and support during running. *Journal of Biomechanics*, 43(14):2709–2716, 2010.
23. H Hatze. A new method for the simultaneous measurement of the moment of inertia, the damping coefficient and the location of the centre of mass of a body segment *in situ*. *European Journal of Applied Physiology*, 34:217–226, 1975.
24. Ernest P Havana. A mathematical model of the human body (Report AMRL-TR-64-102). Technical report, Aerospace Medical Research Laboratory, Ohio, 1964.
25. Georges Hébert. *L'éducation physique, virile et morale par la Méthode Naturelle. Tome I : Exposé doctrinal et principes directeurs de travail*. Vuibert, 1936.
26. Lauren J. Hickox, Blake M. Ashby, and Gordon J. Alderink. Exploration of the validity of the two-dimensional sagittal plane assumption in modeling the standing long jump. *Journal of Biomechanics*, 49(7):1085–1093, 2016.
27. B Hoff and M Arbib. A model of the effects of speed, accuracy, and perturbation on visually guided reaching. *Experimental Brain Research*, 22:285–306, 1992.
28. Katherine R S Holzbaur, Wendy M Murray, and Scott L Delp. A Model of the Upper Extremity for Simulating Musculoskeletal Surgery and Analyzing Neuromuscular Control. *Annals of Biomedical Engineering*, 33(6):829–840, 2005.
29. O. Kanoun, F. Lamiroux, P. B. Wieber, F. Kanehiro, E. Yoshida, and J. P. Laumond. Prioritizing linear equality and inequality systems: Application to local motion planning for redundant robots. In *2009 IEEE International Conference on Robotics and Automation*, pages 2939–2944, 2009.
30. O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.
31. Eirik Kristianslund, Tron Krosshaug, and Antonie J. Van den Bogert. Effect of low pass filtering on joint moments from inverse dynamics: Implications for injury prevention. *Journal of Biomechanics*, 45(4):666–671, 2012.
32. A D Kuo. A least-squares estimation approach to improving the precision of inverse dynamics computations. *Journal of biomechanical engineering*, 120(1):148–59, feb 1998.
33. Mark L. Latash, Stacey Gorniak, and Vladimir M. Zatsiorsky. Hierarchies of Synergies in Human Movements. *Kinesiology*, 40(1):29–38, 2008.
34. Paolo De Leva. Adjustements to Zatsiorsky-Seluyanov's segment inertia parameters. *Journal of Biomechanics*, 29(9):1223–1230, 1996.
35. T.-W. Lu and J.J. OConnor. Bone position estimation from skin marker co-ordinates using global optimisation with joint constraints. *Journal of Biomechanics*, 32(2):129 – 134, 1999.
36. G. Maldonado, H. Bitard, B. Watier, and P. Souères. Evidence of dynamic postural control performance in parkour landing. *Computer Methods in Biomechanics and Biomedical Engineering*, 18(1):1994–1995, 2015.
37. Nicolas Mansard, Olivier Stasse, Paul Evrard, and Abderrahmane Kheddar. A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: the Stack of Tasks. In *ICAR'09: International Conference on Advanced Robotics*, pages 1–6, Munich, Germany, June 2009.

38. Steven T Mccaw, Jacob K Gardner, Lindsay N Stafford, and Michael R Torry. Filtering Ground Reaction Force Data Affects the Calculation and Interpretation of Joint Kinetics and Energetics During Drop Landings. *Journal of Applied Biomechanics*, 29(6):804–809, 2013.
39. Yoshihiko. Nakamura and Yoshihiko. *Advanced robotics : redundancy and optimization*. Addison-Wesley Pub. Co, 1991.
40. D. E. Orin and A. Goswami. Centroidal momentum matrix of a humanoid robot: Structure and properties. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 653–659, Sept 2008.
41. David E. Orin, Ambarish Goswami, and Sung-Hee Lee. Centroidal dynamics of a humanoid robot. *Autonomous Robots*, 35(2):161–176, 2013.
42. L. Saab, N. Mansard, F. Keith, J. Y. Fourquet, and P. Soueres. Generation of dynamic motion for anthropomorphic systems under prioritized equality and inequality constraints. In *2011 IEEE International Conference on Robotics and Automation*, pages 1091–1096, 2011.
43. L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Soueres, and J. Y. Fourquet. Dynamic whole-body motion generation under rigid contacts and other unilateral constraints. *IEEE Transactions on Robotics*, 29(2):346–362, 2013.
44. Claude Samson, Bernard Espiau, and Michel Le Borgne. *Robot control: the task function approach*. Oxford University Press, 1991.
45. John P. Scholz and Gregor Schöner. The uncontrolled manifold concept: Identifying control variables for a functional task. *Experimental Brain Research*, 126(3):289–306, 1999.
46. Ajay Seth, Michael Sherman, Peter Eastman, and Scott Delp. Minimal formulation of joint motion for biomechanisms. *Nonlinear Dynamics*, 62(1):291–303, Oct 2010.
47. B. Siciliano and J. J. E. Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. In *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 1211–1216 vol.2, June 1991.
48. Regan J Standing and Peter S Maulder. A Comparison of the Habitual Landing Strategies from Differing Drop Heights of Parkour Practitioners (Traceurs) and Recreationally Trained Individuals. *Journal of Sports Science and Medicine*, 14(4):723–731, 2015.
49. Olivier Stasse, Thomas Flayols, Rohan Budhiraja, Kevin Giraud-Esclasse, Justin Carpentier, Andrea Del Prete, Philippe Souères, Nicolas Mansard, Florent Lamiraux, Jean-Paul Laumond, Luca Marchionni, Hilario Tome, and Francesco Ferro. TALOS: A new humanoid research platform targeted for industrial applications. working paper or preprint, March 2017.
50. G. Venture, K. Ayusawa, and Y. Nakamura. Motion capture based identification of the human body inertial parameters. In *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4575–4578, 2008.
51. Masaki Wakai and Nicholas P. Linthorne. Optimum take-off angle in the standing long jump. *Human Movement Science*, 24(1):81–96, 2005.
52. David A Winter. *Biomechanics and motor control of human movement, Fourth Edition*. John Wiley & Sons, Inc, 2009.
53. Ge Wu, Sorin Siegler, Paul Allard, Chris Kirtley, Alberto Leardini, Dieter Rosenbaum, Mike Whittle, Darryl D D’Lima, Luca Cristofolini, Hartmut Witte, Oskar Schmid, and Ian Stokes. ISB recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motionPart I: ankle, hip, and spine. *Journal of Biomechanics*, 35(4):543–548, 2002.
54. Ge Wu, Frans C.T. van der Helm, H.E.J. (DirkJan) Veeger, Mohsen Makhsous, Peter Van Roy, Carolyn Anglin, Jochem Nagels, Andrew R. Karduna, Kevin McQuade, Xuguang Wang, Frederick W. Werner, and Bryan Buchholz. ISB recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motionPart II: shoulder, elbow, wrist and hand. *Journal of Biomechanics*, 38(5):981–992, 2005.
55. Vladimir M. Zatsiorsky and V. N. Seluyanov. The mass and inertia characteristics of the main segment of human body. *Biomechanics*, pages 1152–1159, 1983.