IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 December 2022, accepted 1 February 2023, date of publication 10 February 2023, date of current version 15 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3244062

RESEARCH ARTICLE

Human Motion Synthesis Using Trigonometric Splines

ALTAY ZHAKATAYEV[1], (Member, IEEE), NURILLA AVAZOV[2], (Member, IEEE), YURIY ROGOVCHENKO[3], AND MATTHIAS PÄTZOLD[2], (Senior Member, IEEE)

[1]Department of ICT, University of Agder, 4604 Kristiansand, Norway
[2]Department of ICT, University of Agder, 4879 Grimstad, Norway
[3]Department of Mathematics, University of Agder, 4604 Kristiansand, Norway

Corresponding author: Altay Zhakatayev (altay.zhakatayev@uia.no)

This work was supported by the Research Council of Norway under the Carewell Project under Grant 300638.

This work involved human subjects or animals in its research. All ethical and experimental procedures and protocols are in accordance with the CareWell project's Data Management Plan submitted to the Norwegian Centre for Research Data (NSD).

ABSTRACT In this work, we present a simple framework to synthesize human motion. Our main goal is to propose a methodology tailored for inexperienced users to initiate their research in human motion simulation and human motion trajectory optimization. The novelties of the work include the following. First, trigonometric splines are used instead of traditional B-splines to discretize the generalized coordinates and velocities. Second, useful identities for trigonometric splines are derived. Third, this work is intended to be an example, so that even non-skilled users, such as undergraduate students, can perform human motion analysis using a high-level programming language such as MATLAB. Four simulations of human motion are generated: walking, sitting and standing, side-step, and jump. The results of the walking simulation are validated by experiments. Simulation and experimental results are presented and discussed.

INDEX TERMS Human motion synthesis, nonlinear programming problem, trigonometric splines, trajectory optimization.

I. INTRODUCTION

A thorough and deep understanding of biomechanics is essential for healthcare and clinical examination, implementation of safety measures, sports industry, biomedical research, human-robot interaction area, human psychology, space exploration, animation, graphics, gaming industries, and many other fields. Additionally, it seems that humans also accept robots in a society based on how they display elegant behavior [1]. As a result, we still study, investigate and research the nature of human motion.

Human motion research can be broadly divided into experimental and simulation-based research. In experimental research, human subjects perform various activities while their movements are recorded. Motion recording can be performed with numerous motion capture systems, such as the inertial sensor-based XSENS [2], [3], and Rokoko [4], or the visual motion capture systems such as Qualisys [5] and Vicon [6]. The main advantage of the experimental motion recording is the accuracy of the 3D measurements. However, this is mainly true for optical systems, while inertial systems often experience drift in sensor measurements. Experiments are associated with certain disadvantages. One of them is that motion capture experiments are time and resource consuming. Human subjects and their consent are needed to conduct them. Moreover, costly experimental equipment is required to obtain accurate measurements. Another drawback is the noise that is present in any sensor measurement. This in turn requires data post-processing, such as applying filters, smoothing data, removing corrupt data, etc., to extract the required information. The above factors form a bottleneck that impedes the collection of high-quality human motion data.

The advantages of simulation include reduced time, increased data transparency, higher accuracy, absence of noise, and ease of post-processing [7], [8]. Compared to

The associate editor coordinating the review of this manuscript and approving it for publication was Emre Koyuncu.

experimental data, more information can be extracted from simulation data. For instance, it is possible to obtain the knee or ankle torque from simulated motion, but it is difficult or impossible to measure them directly in the experiments. The simulation requires only computational resources and is therefore considered cost-effective. Simulations of human motion can be beneficial in the following: improving the design of prosthetic joints, assisting in surgical procedures to correct gait pathologies, ability to predict novel movement patterns for robots and/or animation characters, increase the robustness to damage and reducing its effects, and synthesis of appropriate gaits for older or younger characters. Simulations of human motion also allow capturing the micro-Doppler signatures of a person for activity recognition and monitoring [9]. Simulation of human motion has also drawbacks. The first one is the reliability of simulation results. Obtaining reliable human body motion simulations is a challenging task due to the multiple degree-of-freedom (DOF) human body model, coupled and nonlinear dynamics, and the complex nature of interactions with the environment. Another drawback is that considerable knowledge (of physics, contact forces, constraints, etc.) and manual efforts are required to obtain good quality simulations. Yet another issue is about the need for a human model with its corresponding mathematical description. This, in turn, places a burden on non-skilled users. In this work, we focus on the simulation of human motion.

There are three ways to simulate human motion: data-driven, optimization-based, and controller-based. Despite significant progress in synthesizing or simulating human motion using optimization tools, the full potential has not yet been realized and there are numerous open questions [10]. For example, problems such as generating personalized subject-specific models, automated synthesis of complex human behaviors, defining good objective functions for optimization problems, and reducing the computational complexity of motion synthesis remain challenging and unsolved [11], [12], [13]. Other challenges include model validation and verification, incorporation of simulation results in the medical industry, increasing simulation accuracy, and accurate computation of muscle forces. Computational human dynamics still lags behind numerical tools used in other industries such as computational fluid dynamics or electromagnetism [8]. Due to the amount of material and sources, and the numerous numerical packages and tools developed for solving optimization problems, it is not easy for beginners to get an overview and initiate their research. Moreover, there are no clear rules or guiding principles for choosing the right numerical tool. This issue is further complicated by the fact that some of the numerical tools (such as SNOPT) are not open source. It often necessitates the integration of open-source packages designed for specific applications to obtain a valid and fully functional framework for human motion synthesis. The reported various human motion synthesis tools are application- and implementation-dependent. These challenges are the reason for the high ''entry cost'' into the field of computational human dynamics. They require knowledge from different domains to synthesize human motions. Our goal is to provide guidance that will allow inexperienced users to quickly and easily synthesize realistic human motion with minimal training. A good summary of the current state of the art in human motion simulation methods can be found in [13]. In this work, we focus on fully synthetic human motion simulations using optimization methods, a powerful approach for the generation of human motion with a minimal amount of input information.

In optimization-based simulations, the polynomial B-splines are commonly used as basis functions to represent unknown variables. The advantages of spline methods include stability, smoothness, continuity, differentiability, and local control [14], [15]. B-splines were used to synthesize the gait motion of an 18-DOF lower human body model in [16], leap, swing, monkey-bar motion of a 22-DOF full human body model in [17], and dynamic break-dancing maneuvers in [12]. Instead of traditional polynomial B-splines, we utilize trigonometric spline basis functions for discretization. We justify it by assuming that human motion is mostly periodic, and so trigonometric functions might be better suited for trajectory representation than B-spline basis functions. As a result, we have developed a technique to synthesize a human motion using the trigonometric spline method. We show in this work that the trigonometric spline method can simulate human motion as efficiently as the B-spline method.

The major contributions of this paper are:

- Trigonometric spline method is used to discretize the generalized coordinates and velocities. New expressions for time derivatives of the trigonometric splines are derived. To the best knowledge of the authors, this is the first time trigonometric splines are utilized for human motion synthesis.
- We demonstrate that human motion can be synthesized using robust and high-level programming languages such as MATLAB, which is easy to code and debug, but were previously considered rather slow. This is possible thanks to efficient sensitivity analysis algorithms.
- This work is aimed to be an example of human motion synthesis problem with the intent to lower the entry barrier for others into this research field.

The paper is organized as follows. Section II provides a review of the existing literature on human motion simulations. The nonlinear programming problem, its objective function and constraints are described in Sec. III, IV, and V, respectively. Four simulation examples are presented in Sec. VI, while the performed experiments are described in Sec. VII. The results of both simulations and experiments are discussed in Sec. VIII.

II. RELATED WORKS

In data-driven human motion simulations, the recorded human motion is either visualized or modified to generate

a new motion using blending, optimization, or interpolation techniques. The resulting problems are also called *tracking problems* in the optimization community, where the aim is to obtain the simulated motion as close as possible to the experimentally recorded motion. With the advent of reinforced learning techniques (neural networks), great advances have been made in data-driven simulations of human motion. Examples include synthesis of human motion from a large set of training data without manual preprocessing or human intervention [18], a hierarchical control framework for environment-aware locomotion based on limited training data [19], a physics-based framework for animating figures that combines goal-directed reinforcement learning with data [20]. Optimization methods are also used in synthesizing human motion from experimental data. For example, various dynamic human motions were generated from reference animations using optimization methods and empirical models for the behavior of human limbs' momentum [21]. Optimal feedback control was developed to simulate reference human motions subject to disturbances and environmental changes [22]. Other examples of data-driven motion control include the use of feedback error learning [23], support vector machines [24], curriculum learning [25], and motion synthesis from textual descriptions [26] and audio [27]. Context-specific locomotion steps were synthesized using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm and the full-body inverse kinematics in [28]. A new and interesting extension in this direction was the utilization of videos to capture human motion, i.e., human 3D trajectories were inferred from videos [29]. A human dynamic model with 48 DOF was used to visualize and simulate human motion based on experimental motion capture data [30]. Multi-character simulations were also obtained from motion capture data using reinforcement learning [31], [32] and hierarchical reinforcement learning [33]. Data-driven human motion simulations are popular in computer graphics, gaming, and the machine learning community, which can be explained in part by the presence of model-free deep reinforcement learning techniques [19]. In other words, there are data-driven human motion simulation methods that do not require a human model. However, this group of methods have drawbacks, such as lacking or limited predictive power (compared to optimization problems, since most motion editing tools try to preserve the original motion), producing erroneous, i.e., physically incorrect and jerky motions, and the inability to accurately model force interactions with the environment [34]. However, recent developments in reinforcement learning and deep reinforcement learning based motion synthesis, which does not require motion capture data, are very promising [35]. Our work can be considered as complementary to their work.

Multibody dynamics is categorized as forward dynamics (kinematic variables are unknowns) and inverse dynamics (force/torque variables are unknowns). In reality, neither forces/torques nor positions and velocities are available before performing a simulation of the human body motion. Therefore, these problems fall into the optimization problem category. The problems in this category are called *predictive dynamics* problems [7] (also called *trajectory optimization*, *motion synthesis* or *motion control* in different engineering fields). The first application of the optimization problem to synthesize human motion was presented in [36]. Multiple techniques are available for solving an optimization problem [37], [38]. The optimal control methods employed to model a human motion are summarized in [39] and [10]. Mathematical parameterization of the biomechanical experimental human walking data was used to develop a human walking model [40]. In essence, the experimental kinematic data of human joints without dynamic equations were used to construct the dynamic human model. A walking simulation of the human lower body musculoskeletal model with 23 DOF was presented in [41]. In [42], a model for generating style- and subject-specific human motion was presented based on optimizing the parameters of the physical model (human model with 35 DOF) and their subsequent synthesis. The synthesis of the walking motion of a biped robot with 13 DOF and a lower human body skeletal model with 18 DOF were presented in [16] and [43], respectively. Gait simulation for a full-body skeletal model was synthesized in [44]. Human gait was generated using fatigue as a cost function in the optimization problem for a 9 DOF model [45]. Other types of human motion synthesized using optimization tools include jumping [46], [47], [48], lifting [7], pedaling [49], standing up, headspin, handspin, push-ups and crawling [12]. An interesting contact-invariant optimization method that optimizes both the contact events and the motion trajectory was presented in [50]. In this work, the optimization problem is solved directly for the contact points instead of manually specifying the contact information or solving it indirectly. However, to achieve a favorable trade-off between physical realism and the optimization efficiency, a simplified human model was utilized. A framework for the generation of locomotion strategies for bipedal creatures was proposed in [51]. Different dynamic behaviors using the 41 DOF human body model and a combination of simple objective functions were synthesized in [12]. The novelty of this work was the utilization of optimization windows: a full trajectory optimization would be prohibitively expensive, so the authors divided the simulation time into multiple windows and performed the optimization sequentially over each window. The advantages of the optimization-based motion synthesis include minimum required input (no experimental data), full flexibility (new and unique motions can be predicted), and variability (different types of motion can be generated). However, its weaknesses are the need for a mathematical model of a human body and its interaction with the environment.

While early work focused primarily on the offline synthesis of simple (and mostly cyclic) human motions within a short period of time, such as a phase of walking or jumping, later

research focused more on the online synthesis of interactive and/or more complex motions over a longer period of time. These methods can also be referred to as controller-based human motion synthesis, where the task is to design a controller that allows a human avatar (model) to simulate a specific motion in real time. The obtained interactive motion synthesis is often used in character animation. One common feature of these controller-based methods is that they often rely on a simplified human model (inverted pendulum) and use offline reference motion generators. A physics-based interactive framework for human motion synthesis was presented in [52]. The advantages of the method include minimal physical parameter tuning and an adaptive controller for an arbitrary character model. However, the proposed optimization problem does not enforce dynamics at the joints or joint torque constraints, and the naturalness of the resulting motion is achieved by enforcing a slow change in joint torques in time. In [53], a control strategy for humanoid walking movements was presented that can be generalized across gait parameters, movement styles, and character proportions. The control strategy is based on a simplified inverted pendulum model. A low-dimensional spring-loaded inverted pendulum model and a two-stage optimization process were used to simplify the synthesis of the highly-dynamic human gait [11], [54]. In the first (step) optimization stage, the simplified pendulum model is used to calculate the step length and heading. While in the second (torque) optimization stage, the optimal torque values are found for a full 37 DOF human model. Human gait was generated by considering the influence of leg mechanics on human motor control, which is another form of state feedback control [55]. However, as the authors note, this method lacks predictive power and can be used to generate mostly repetitive motions such as gait. Walking patterns at different speeds for a human body model with 30 DOF were generated in [56]. The authors proposed a novel, biologically inspired state feedback control in which muscle excitation is based on the muscle force and length, and other state values. In other words, joint torques depend on the system state. A sampling-based algorithm was used to generate collision-free and balanced whole-body trajectories for a 38 DOF NASA Valkyrie robot [57]. Interestingly, the inverse kinematics calculations dominated over 95% of the computation time. Reinforcement learning was used to generate diverse terrain traversing motions of the 28 DOF humanoid model in [58]. It was demonstrated that simple and generic reward functions combined with training over a wide range of environmental conditions can lead to the development of non-trivial locomotion skills. Interesting work was presented in [34], where for a given sequence of high-level tasks, a human motion was synthesized from motion primitives using linear temporal logic specifications. However, the proposed method is efficient at generating complex motions if motion primitives are available, but it cannot generate motion primitives themselves. The main shortcomings of controller-based methods are the following. They often use a simplified human model (especially for foot placement calculations). Moreover, these methods rely on forward dynamics simulators, such as *Open Dynamics Engine*.

III. OPTIMAL CONTROL AND NONLINEAR PROGRAMMING PROBLEMS

In this work, we will not specify the dynamics of the human body, because there are numerous formulations (Newton-Euler, Lagrangian, Hamiltonian, spatial notation, Kane's equations, etc.). Instead, we leave the reader freedom to choose the specific formulation and describe a model-free optimization. However, for clarity, we introduce the inverse dynamics equations of an arbitrary mechanical system with n_q DOF as follows

$$M(\mathbf{q})\ddot{\mathbf{q}} + K(\mathbf{q}, \dot{\mathbf{q}}) + W(\mathbf{q}) = \mathbf{Q}, \tag{1}$$

where $\mathbf{q} = [q_1\ q_2\ \dots q_{n_q}]^T \in \mathbb{R}^{n_q}$ and $\dot{\mathbf{q}} = [\dot{q}_1\ \dot{q}_2\ \dots \dot{q}_{n_q}]^T \in \mathbb{R}^{n_q}$ are the vectors of the generalized coordinates and generalized velocities, respectively; $M(\mathbf{q}) \in \mathbb{R}^{n_q \times n_q}$ is the inertia matrix; $K(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n_q}$ is the vector of the Coriolis and normal inertial forces; $W(\mathbf{q}) \in \mathbb{R}^{n_q}$ is the vector of gravitational forces, and $\mathbf{Q} = [Q_1\ Q_2\ \dots Q_{n_q}]^T \in \mathbb{R}^{n_q}$ is the vector of the generalized forces. The superscript $()^T$ means the transpose operator. The 43-DOF human model with 20 links is used to generate the human motion in this work, see Fig. 1. The global coordinate system has unit vectors x_0-y_0-z_0. We emphasize that the specific kinematic and dynamic descriptions, as well as the sensitivity analysis of the model, are not given so that the paper and the reader do not depend or rely on them.

The synthesis of human motion is a trajectory optimization problem and can be solved by either an *optimal control problem* or a *nonlinear programming problem*. The optimal control problem (OCP) is formulated as follows

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{u}}\ \mathcal{J}(\mathbf{x}, \mathbf{u}) \\ &\text{s.t. } \mathcal{G}(\mathbf{x}, \mathbf{u}) \le 0 \\ &\quad \mathcal{H}(\mathbf{x}, \mathbf{u}) = 0 \\ &\quad \dot{\mathbf{x}} = \mathcal{F}(\mathbf{x}, \mathbf{u}). \end{aligned} \tag{2}$$

where $\mathcal{J}$ is the *objective function*, also called the *cost function*, which is a scalar function that needs to be minimized. $\mathcal{G}$ is a set of nonlinear inequality constraints, $\mathcal{H}$ is a set of nonlinear equality constraints, $\mathcal{F}$ is the state equation that describes the dynamics of the system as a set of first-order nonlinear differential equations, while $\mathbf{x}$ and $\mathbf{u}$ are the state and control vectors, respectively. Usually, $\mathbf{x} = [\mathbf{q}^T\ \dot{\mathbf{q}}^T]^T \in \mathbb{R}^{2n_q}$ and $\mathbf{u} = \mathbf{Q}$ from (1). Solving the OCP is a tedious task, especially for complex systems in real life. Roughly speaking, there are two methods to solve the OCP: analytical and numerical [59]. Numeric methods can be further divided into direct and indirect methods[1] [38], [61], [62]. Indirect methods are based on

[1]There is also a dynamic programming method, but it is not practical for systems with many DOFs [60].

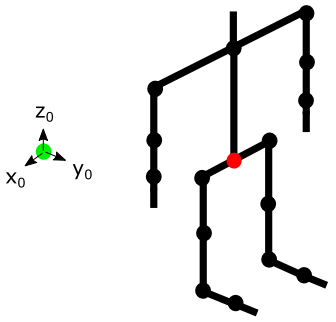


FIGURE 1. Schematic diagram of a humanoid model with 43 DOF and 20 links. The green dot is the origin of the global frame of reference x_0-y_0-z_0, the red dot is the pelvis point, black dots are joints.

the Pontryagin minimum principle and convert the OCP into a two-point boundary value problem. Indirect methods can also be referred to as "optimization then discretization" methods [63]. Direct methods are based on the Karush-Kuhn-Tucker conditions and convert the OCP into an NLP. Direct methods are also referred to as "discretization then optimization" methods. The OCP can be considered as an infinite-dimensional optimization problem, while the NLP is the finite-dimensional version of the OCP [38]. The NLP can be defined as

$$\begin{aligned} &\min_{\mathbf{y}} \ \mathcal{J}(\mathbf{y}) \\ &\text{s.t. } \mathcal{G}(\mathbf{y}) \leq 0 \\ &\quad \mathcal{H}(\mathbf{y}) = 0. \end{aligned} \tag{3}$$

For the NLP, $\mathbf{y}$ is the vector of decision variables that is different from the vector of state variables $\mathbf{x}$ in the OCP (2). There are three ways to convert the OCP into the NLP: numerical integration, collocation, and differential inclusion [37], [64]. In the numerical integration methods, the control input vector $\mathbf{u}$ is discretized and selected as a decision variable vector $\mathbf{y}$, while the state vector $\mathbf{x}$ is found by integrating the dynamic equations [46]. In the collocation methods, both the state vector $\mathbf{x}$ and the control input vector $\mathbf{u}$ are discretized and chosen as the decision variables [49]. In the differential inclusion methods, the discretized state vector $\mathbf{x}$ is used as the decision variable vector $\mathbf{y}$, while the control input vector $\mathbf{u}$ is found by imposing the dynamic equations (1), [44]. It should be noted that the shooting methods, which involve the integration of the dynamic equations of motion, are in general more costly than the collocation or the differential inclusion methods [48]. Since we use inverse dynamics in this work, the differential inclusion method is chosen for the NLP. An efficient solution of the optimization problem with gradient-based methods depends largely on the objective function and constraint gradient information. The fast calculation of gradients makes the solution of optimization problems feasible even for the large-DOF systems.

A. DISCRETIZATION

To convert the OCP into an NLP using the differential inclusion method, we must discretize the generalized coordinates and its time derivatives. This is accomplished by using the *spline basis functions* $\mathcal{N}_{l,\gamma}(t)$ of degree γ as

$$\begin{aligned} q_i(t_j) &= \sum_{l=0}^{N} \mathcal{N}_{l,\gamma}(t_j) y_{i,l}, \quad \dot{q}_i(t_j) = \sum_{l=0}^{N} \dot{\mathcal{N}}_{l,\gamma}(t_j) y_{i,l} \\ \ddot{q}_i(t_j) &= \sum_{l=0}^{N} \ddot{\mathcal{N}}_{l,\gamma}(t_j) y_{i,l}, \end{aligned} \tag{4}$$

where $y_{i,l}$ $(l = 0, \ldots, N)$ are control points used to discretize the generalized coordinate i; $N+1$ is the number of control points per DOF (this is also equal to the number of basis functions), and t_j is the specific time step. $\dot{\mathcal{N}}_{l,\gamma}(t_j)$ and $\ddot{\mathcal{N}}_{l,\gamma}(t_j)$ are, respectively, the first and the second-order time derivatives of the spline basis function $(d\mathcal{N}_{l,\gamma}(t)/dt$ and $d^2\mathcal{N}_{l,\gamma}(t)/dt^2)$ evaluated at t_j. The so-called *knot vector* is formed from the sampling time steps $t_j, j = 0, \ldots, n_t$, where n_t is the number of sampling intervals. Some of the terms of the knot vector might have a multiplicity higher than 1. The total number of control points for the entire dynamic system is $n_q(N+1)$. The control points are the optimization variables. There is a relation between the number of control points, the degree of basis functions and the dimension of the knot vector l_{kv}: $l_{kv} - 2 = N + \gamma$. Please note that $l_{kv} \neq n_t + 1$. If the simulation duration is t_{sim} and the discretization is uniform with the number of sampling intervals n_t, then $\Delta t = t_{sim}/n_t$ and $t_j = j\Delta t, j = 0, \ldots, n_t$.

B. TRIGONOMETRIC SPLINES

Trigonometric splines were introduced in [65] and later useful identities for them were derived [66], [67]. The difference between the B-splines and the trigonometric splines is that in the former the basis functions are polynomials, while in the latter the basis functions are trigonometric functions. Human motion is mostly a periodic motion. Therefore, we assume that use of the trigonometric splines is more suitable than polynomial B-splines for the discretization of human motion.

Trigonometric splines are defined similarly to polynomial splines by recursive Cox-de Boor relations as

$$\begin{aligned} \mathcal{N}_{l,0}(t) &= \begin{cases} 1 & \text{for } t_l \leq t < t_{l+1} \\ 0 & \text{otherwise} \end{cases} \\ \mathcal{N}_{l,\gamma}(t) &= \frac{\sin(t - t_l)}{\sin(t_{l+\gamma} - t_l)} \mathcal{N}_{l,\gamma-1}(t) \\ &\quad + \frac{\sin(t_{l+\gamma+1} - t)}{\sin(t_{l+\gamma+1} - t_{l+1})} \mathcal{N}_{l+1,\gamma-1}(t) \end{aligned} \tag{5}$$

with the restriction that $t < \pi$. In case $t \geq \pi$, t can always be re-scaled to satisfy this condition. In special cases, the following conditions are applied

$$\begin{aligned} &\mathcal{N}_{l,\gamma}(t) \\ &= \begin{cases} \frac{\sin(t_{l+\gamma+1} - t)}{\sin(t_{l+\gamma+1} - t_{l+1})} \mathcal{N}_{l+1,\gamma-1}(t) \text{ if } \sin(t_{l+\gamma} - t_l) = 0 \\ \frac{\sin(t - t_l)}{\sin(t_{l+\gamma} - t_l)} \mathcal{N}_{l,\gamma-1}(t) \text{ if } \sin(t_{l+\gamma+1} - t_{l+1}) = 0 \\ 0 \quad \text{if both denominators are } 0. \end{cases} \end{aligned} \tag{6}$$

Time derivatives of basis spline functions can also be found by recursive relations

$$\dot{\mathcal{N}}_{l,\gamma}(t) = \gamma\Big(\frac{\cos(t-t_l)}{\sin(t_{l+\gamma}-t_l)}\mathcal{N}_{l,\gamma-1}(t) - \frac{\cos(t_{l+\gamma+1}-t)}{\sin(t_{l+\gamma+1}-t_{l+1})}\mathcal{N}_{l+1,\gamma-1}(t)\Big). \tag{7}$$

$$\begin{aligned}\ddot{\mathcal{N}}_{l,\gamma}(t) = &-\gamma\Big(\frac{\sin(t-t_l)}{\sin(t_{l+\gamma}-t_l)}\mathcal{N}_{l,\gamma-1}(t) + \frac{\sin(t_{l+\gamma+1}-t)}{\sin(t_{l+\gamma+1}-t_{l+1})}\mathcal{N}_{l+1,\gamma-1}(t)\Big)\\ &+\gamma(\gamma-1)\Big(\frac{\cos^2(t-t_l)}{\sin(t_{l+\gamma}-t_l)\sin(t_{l+\gamma-1}-t_l)}\times\mathcal{N}_{l,\gamma-2}(t)\\ &-\frac{\cos(t-t_l)\cos(t_{l+\gamma}-t)}{\sin(t_{l+\gamma}-t_l)\sin(t_{l+\gamma}-t_{l+1})}\mathcal{N}_{l+1,\gamma-2}(t)\\ &-\frac{\cos(t-t_{l+1})\cos(t_{l+\gamma+1}-t)}{\sin(t_{l+\gamma}-t_{l+1})\sin(t_{l+\gamma+1}-t_{l+1})}\mathcal{N}_{l+1,\gamma-2}(t)\\ &+\frac{\cos^2(t_{l+\gamma+1}-t)}{\sin(t_{l+\gamma+1}-t_{l+1})\sin(t_{l+\gamma+1}-t_{l+2})}\\ &\times\mathcal{N}_{l+2,\gamma-2}(t)\Big).\end{aligned} \tag{8}$$

If any of the denominators in (7) and (8) is zero, then the corresponding term is assigned to be zero. The examples of the trigonometric basis spline functions of a different degree and their time derivatives, defined over the knot vector $[0\ 1\ldots 8]$,[2] are shown in Fig. 2. The equations (5)-(8) are substituted into (4) to compute the generalized coordinates, velocities, and accelerations, which in turn are then substituted into (1) to transform a set of nonlinear differential equations into a set of nonlinear algebraic equations.

C. GRADIENT AND JACOBIAN MATRICES

For further analysis we will need the Jacobian matrices of the generalized coordinates $\mathbb{J}_{q,y} \in \mathbb{R}^{n_q \times n_q(N+1)}$ with respect to the optimization variables evaluated at specific time steps. Each entry of the Jacobian matrix is $\mathbb{J}_{q,y}(i,j) = \frac{\partial q_i}{\partial y_{k,l}}$, where the indexes k and l satisfy the following identity $(k-1)(N+1)+l+1=j$.

$$\mathbb{J}_{q,y}(t_j) = \begin{bmatrix}\mathbf{n} & \mathbf{0} & \ldots & \mathbf{0}\\ \mathbf{0} & \mathbf{n} & \ldots & \mathbf{0}\\ & & \ldots & \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{n}\end{bmatrix}, \tag{9}$$

where $\mathbf{n} = [\mathcal{N}_{0,\gamma}(t_j)\ \mathcal{N}_{1,\gamma}(t_j)\ldots\mathcal{N}_{N,\gamma}(t_j)] \in \mathbb{R}^{1\times(N+1)}$ is the vector of the basis function values at a specific time, while $\mathbf{0} = [0\ 0\ldots 0] \in \mathbb{R}^{1\times(N+1)}$. Similarly, $\mathbb{J}_{\dot{q},y} \in \mathbb{R}^{n_q\times n_q(N+1)}$ and $\mathbb{J}_{\ddot{q},y} \in \mathbb{R}^{n_q\times n_q(N+1)}$ define the Jacobian matrices of the generalized velocities and accelerations with respect to the optimization variables, correspondingly. These matrices are

[2]The knot vector was scaled to satisfy the $t < \pi$ condition.

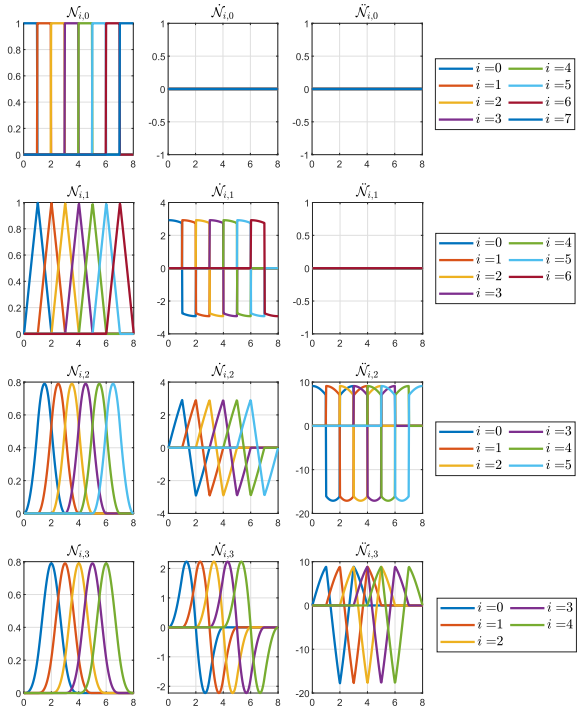


FIGURE 2. Trigonometric spline basis functions and their time derivatives for different degrees $\gamma = 0, 1, 2, 3$ and over the knot vector $[0\ 1\ldots 8]$.

constructed as

$$\mathbb{J}_{\dot{q},y}(t_j) = \begin{bmatrix}\mathbf{v} & \mathbf{0} & \ldots & \mathbf{0}\\ \mathbf{0} & \mathbf{v} & \ldots & \mathbf{0}\\ & & \ldots & \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{v}\end{bmatrix},\ \mathbb{J}_{\ddot{q},y}(t_j) = \begin{bmatrix}\mathbf{a} & \mathbf{0} & \ldots & \mathbf{0}\\ \mathbf{0} & \mathbf{a} & \ldots & \mathbf{0}\\ & & \ldots & \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{a}\end{bmatrix}, \tag{10}$$

where the vector $\mathbf{v} = [\dot{\mathcal{N}}_{0,\gamma}(t_j)\ \ \dot{\mathcal{N}}_{1,\gamma}(t_j)\ldots\dot{\mathcal{N}}_{N,\gamma}(t_j)] \in \mathbb{R}^{1\times(N+1)}$ and the vector $\mathbf{a} = [\ddot{\mathcal{N}}_{0,\gamma}(t_j)\ \ \ddot{\mathcal{N}}_{1,\gamma}(t_j)\ldots\ddot{\mathcal{N}}_{N,\gamma}(t_j)] \in \mathbb{R}^{1\times(N+1)}$. These matrices will be helpful to evaluate the gradients of the objective and the constraint functions. Additionally, these Jacobian matrices can be used to rewrite (4) in the following form

$$\mathbf{q}(t_j) = \mathbb{J}_{q,y}(t_j)\mathbf{y},\ \ \dot{\mathbf{q}}(t_j) = \mathbb{J}_{\dot{q},y}(t_j)\mathbf{y},\ \ \ddot{\mathbf{q}}(t_j) = \mathbb{J}_{\ddot{q},y}(t_j)\mathbf{y}, \tag{11}$$

where $\mathbf{q}(t_j)$ is the vector of the generalized coordinates evaluated at time t_j, while $\mathbf{y} = [y_{1,0}\ y_{1,2}\ldots y_{1,N}\ y_{2,0}\ y_{2,1}\ldots y_{2,N}\ldots y_{n_q,N}]^T \in \mathbb{R}^{n_q(N+1)\times 1}$ is the vector of the optimization variables.

Given an arbitrary scalar function of the generalized coordinates, velocities and accelerations $f(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$, its gradient matrices are

$$\mathbb{J}_{f,q} = \frac{\partial f}{\partial \mathbf{q}},\ \ \mathbb{J}_{f,\dot{q}} = \frac{\partial f}{\partial \dot{\mathbf{q}}},\ \ \mathbb{J}_{f,\ddot{q}} = \frac{\partial f}{\partial \ddot{\mathbf{q}}},\ \text{all} \in \mathbb{R}^{1\times n_q}. \tag{12}$$

The derivative with respect to an optimization variable $y_{k,l}$ can be found as

$$\frac{\partial f}{\partial y_{k,l}} = \sum_{i=1}^{n_q}\frac{\partial f}{\partial q_i}\frac{\partial q_i}{\partial y_{k,l}} + \frac{\partial f}{\partial \dot{q}_i}\frac{\partial \dot{q}_i}{\partial y_{k,l}} + \frac{\partial f}{\partial \ddot{q}_i}\frac{\partial \ddot{q}_i}{\partial y_{k,l}}. \tag{13}$$

The corresponding gradient $\mathbb{J}_{f,y} \in \mathbb{R}^{1\times n_q(N+1)}$ can be found as

$$\mathbb{J}_{f,y} = \mathbb{J}_{f,q}\mathbb{J}_{q,y} + \mathbb{J}_{f,\dot{q}}\mathbb{J}_{\dot{q},y} + \mathbb{J}_{f,\ddot{q}}\mathbb{J}_{\ddot{q},y}. \tag{14}$$

Similarly, the Jacobian matrix of an arbitrary vector-valued function f is found from (14).

IV. OBJECTIVE FUNCTION

As the objective function in (3), we have chosen the commonly used dynamic effort, which is time integral of all joint torques (1) squared

$$\mathcal{J}(\mathbf{y}) = \sum_{j=0}^{n_t} \frac{1}{\mathbf{Q}_{max}^2}\mathbf{Q}(t_j)^T\mathcal{W}\mathbf{Q}(t_j), \tag{15}$$

where $\mathcal{W} \in \mathbb{R}^{n_q\times n_q}$ is the diagonal weight matrix; $\mathbf{Q}_{\max} \in \mathbb{R}$ is the maximum value of the generalized force vector, and $\mathbf{Q}(t_j)$ is the torque vector (1) evaluated at time t_j. Since the torques depend on the generalized coordinates, velocities, and accelerations, which in turn depend on the optimization variables (4), the cost function is a function of $\mathbf{y}$. Recently, it has been shown that even the arbitrary whole-body movements are chosen based on minimizing energy cost [68]. The gradient of the cost function at each sampling time is calculated as

$$\mathbb{J}_{\mathcal{J},y} = \mathbb{J}_{\mathcal{J},Q}(\mathbb{J}_{Q,q}\mathbb{J}_{q,y} + \mathbb{J}_{Q,\dot{q}}\mathbb{J}_{\dot{q},y} + \mathbb{J}_{Q,\ddot{q}}\mathbb{J}_{\ddot{q},y}). \tag{16}$$

V. CONSTRAINTS

There are two types of constraints: equality and inequality constraints. The constraints used in (3) will be described below.

A. INITIAL AND FINAL CONDITIONS

As mentioned above, our optimization variables are the control points $y_{i,l}$, $i = 1,\ldots,n_q$, $l = 0,\ldots,N$. While the generalized coordinates and velocities are the variables in (1). Therefore, the initial and final conditions for the NLP variables $\mathbf{y}$ are taken from the values of the vectors $\mathbf{q}$ and $\dot{\mathbf{q}}$ at the initial ($\mathbf{q}(0)$ and $\dot{\mathbf{q}}(0)$) and final times ($\mathbf{q}(t_{sim})$ and $\dot{\mathbf{q}}(t_{sim})$) by solving the following equations

$$\begin{aligned} \mathbf{q}(0) &= \mathbb{J}_{q,y}(0)\mathbf{y}, \quad \dot{\mathbf{q}}(0) = \mathbb{J}_{\dot{q},y}(0)\mathbf{y} \\ \mathbf{q}(t_{sim}) &= \mathbb{J}_{q,y}(t_{sim})\mathbf{y}, \quad \dot{\mathbf{q}}(t_{sim}) = \mathbb{J}_{\dot{q},y}(t_{sim})\mathbf{y}. \end{aligned} \tag{17}$$

B. CONSTRAINTS ON GENERALIZED COORDINATES AND VELOCITIES

Another type of constraint is imposed on the generalized coordinates and velocities. These constraints can be written as

$$\mathbf{q}_l \le \mathbf{q} \le \mathbf{q}_u, \quad \dot{\mathbf{q}}_l \le \dot{\mathbf{q}} \le \dot{\mathbf{q}}_u, \tag{18}$$

where $\mathbf{q}_l$ ($\mathbf{q}_u$) and $\dot{\mathbf{q}}_l$ ($\dot{\mathbf{q}}_u$) are the corresponding lower (upper) boundaries of the generalized coordinates and velocities. The constraints (18) are imposed at the arbitrary time t_j, and the lower and upper boundary vectors can also vary with time.

C. CONSTRAINTS ON TORQUES

The constraints on the torques are the following

$$\mathbf{Q}_l \le \mathbf{Q} \le \mathbf{Q}_u, \tag{19}$$

where $\mathbf{Q}_l$ and $\mathbf{Q}_u$ are the lower and upper boundaries imposed on the torque vector. $\mathbf{Q}_u$ can be used as $\mathbf{Q}_{\max}$ in the objective function (15).

D. PELVIS VELOCITY CONSTRAINT

The global translational velocity of the pelvis is defined as ${}^p\mathbf{v}_g \in \mathbb{R}^{3\times 1}$ and is equivalent to the translational velocity of the human body as a whole. Given the step length l_{step} and the step time t_{step}; its modulus can be found as $|{}^p\mathbf{v}_g| = l_{step}/t_{step}$. The constraint on the pelvis velocity is the following

$${}^p\mathbf{v}_{g,l} \le |{}^p\mathbf{v}_g| \le {}^p\mathbf{v}_{g,u}, \tag{20}$$

here ${}^p\mathbf{v}_{g,l}$ and ${}^p\mathbf{v}_{g,u}$ are, respectively, the lower and upper boundaries of the pelvis velocity.

E. FOOT-GROUND PENETRATION CONSTRAINT

An important aspect of human motion synthesis is the foot-ground interaction modeling. Foot-ground interaction is responsible for ensuring and restoring the dynamic balance, keeping the translational motion of the body, and its stability [43]. The simplest model for foot-ground interaction is to model a condition where a flat foot touches the ground with its entire surface. However, this model does not capture the actual dynamics of foot-ground interaction. More complex models exist, like spring-damper models and *foot contact points*. The foot-ground interaction in [43] was not modeled by contact points, but by using conditions for generalized coordinates. Sometimes the foot-ground interaction is modeled by a spring-damper system [41], [45], [47], [48], other times it is modeled by diverse friction models, such as the Coulomb friction model [17], [52]. In this work, we use a simple, yet powerful foot-ground interaction model based on the foot contact point (FCP) and the foot-ground interaction matrix. We note that when a human motion is simulated, then a foot-ground interaction model is required. However, when a human motion is synthesized, then a foot-ground interaction is imposed through constraints.

Let the number of FCPs per foot be denoted as n_f, then their total number for the human model is $2n_f$. For instance, $n_f = 1$ in [42], $n_f = 3$ in [30], $n_f = 4$ in [29], $n_f = 5$ in [47], $n_f = 8$ in [48], $n_f = 10$ in [45]. In our model $n_f = 6$, as shown in Fig. 3. The position vectors of the FCP, denoted as ${}^o\mathbf{r}_j \in \mathbb{R}^{3\times 1}, j = 1,\ldots,n_f$, are selected in the local frame of reference of the foot. The contact points $j = 1,\ldots,4$ are located in the local coordinate system of the right and left foots. The remaining contact points $j = 5, 6$ are in the local reference frame of the right and left toes. Also, the FCPs 1-2 describe the heel of the foot, the points 3-4 depict the joint between foot and toes, and finally the points 5-6 denote the toes.

The global z coordinates of the FCPs are denoted as ${}^oz_{g,j} \in \mathbb{R}$, $j = 1,\ldots,n_f$. The no-ground penetration

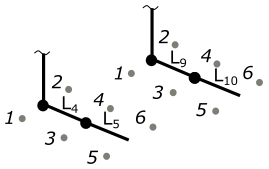


FIGURE 3. Location of the foot contact points.

constraint, which imposes the condition that the FCPs cannot penetrate the ground, can be written as

$$0 \leq \mathcal{Z}_g, \tag{21}$$

where $\mathcal{Z}_g = [{}^o z_{g,1}, \ldots, {}^o z_{g,n_f}]^T \in \mathbb{R}^{n_f \times 1}$.

F. FOOT CONTACT CONSTRAINT

The constraints which impose the specific positions of the FCP on the ground (where the foot touches the ground) need to be specified. Let us denote the local and the global positions of the FCP as ${}^o\mathbf{r}_j \in \mathbb{R}^{3\times 1}$ and ${}^o\mathbf{r}_{g,j} \in \mathbb{R}^{3\times 1}$, $j = 1, \ldots, n_f$, respectively. Then, these constraints can be defined as

$$\mathcal{R}_g = \mathcal{R}_g^n, \tag{22}$$

where $\mathcal{R}_g = [{}^o\mathbf{r}_{g,1} \ldots {}^o\mathbf{r}_{g,n_f}] \in \mathbb{R}^{3\times n_f}$ is the matrix containing the radius vectors of the FCP in the global frame, while $\mathcal{R}_g^n \in \mathbb{R}^{3\times n_f}$ is the matrix specifying the numeric values of the points where the foot touches the ground. Similar constraints apply to the time derivatives of the FCPs as

$$\mathcal{V}_g = 0, \tag{23}$$

where $\mathcal{V}_g = [{}^o\mathbf{v}_{g,1} \ldots {}^o\mathbf{v}_{g,n_f}] \in \mathbb{R}^{3\times n_f}$ is the translational velocity of the foot contact points in the global frame. Equation (23) imposes the no-slip boundary condition on the FCPs with the ground.

Conditions (22) and (23) are valid for all FCPs (for each foot) and are applied at every sampling time. However, these constraints cannot describe the situations where only a subset of FCPs is touching the ground. For example, when a heel touches the ground, or a toe lifts off the ground. To describe these situations, we introduce the *foot contact constraint matrix* $\Lambda \in \mathbb{R}^{n_f \times n_t}$: $\Lambda(i, j) = 1$ if a foot contact point i touches the ground at sampling time t_j, otherwise the element is zero. For each foot there is a single matrix Λ, so for a human model, there are two foot contact matrices. Using Λ, more complex situations of the foot-ground interaction can be modeled. Specifically, if at the current sampling time t_j the foot contact point i has a contact with the ground, then the ground contact (22) and no-slip (23) boundary conditions are applied for that point.

G. ZERO-MOMENT POINT CONSTRAINT

The next constraint that we consider is the *zero moment point* constraint. This constraint is widely used in humanoid and biped robot motion control, stabilization, and planning applications. The zero-moment point (ZMP) is the point on the ground around which the component of the net reaction torque acting on the body along one of the horizontal axes is zero. In short, this is the center of pressure of the footprints. This point is responsible for the dynamic stability of the movement. If it is located within the foot support region, then the human posture and movement are stable, otherwise, the body could fall. We decided to use the ZMP calculation from [44].

Firstly, we need to find the global generalized forces at the pelvis. Let us denote the global force and torque at the pelvis with respect to the global coordinate system as $\mathcal{F}_p \in \mathbb{R}^{3\times 1}$ and $\mathcal{T}_p \in \mathbb{R}^{3\times 1}$, respectively. Secondly, the global forces $\mathcal{F}_o \in \mathbb{R}^{3\times 1}$ and torques $\mathcal{T}_o \in \mathbb{R}^{3\times 1}$ at the origin can be found by transferring the global forces and torques from the pelvis to the origin as $\mathcal{F}_o = -\mathcal{F}_p$ and $\mathcal{T}_o = -\mathcal{T}_p - ({}^p\mathbf{r}_g \times \mathcal{F}_p)$, where ${}^p\mathbf{r}_g \in \mathbb{R}^{3\times 1}$ is the global position of the pelvis, and $\times$ is the cross product. Finally, the ZMP position ${}^{zmp}\mathbf{r}_g$ in the global frame can be found as

$${}^{zmp}\mathbf{r}_g = [-\frac{\mathcal{T}_o[2]}{\mathcal{F}_o[3]} \quad \frac{\mathcal{T}_o[1]}{\mathcal{F}_o[3]} \quad 0]^T. \tag{24}$$

This equation expresses the condition that the x_0 and y_0 components of the global reaction torque acting on the human model are zero. The condition that the ZMP is located within the foot support polygon can be expressed as

$$\varepsilon\Big(({}^{zmp}\mathbf{r}_g - {}^o\mathbf{r}_{g,i}) \times ({}^o\mathbf{r}_{g,j} - {}^o\mathbf{r}_{g,i})\Big) \cdot z_0 \leq 0, \quad i = 1, \ldots, n_p \tag{25}$$

where ${}^{zmp}\mathbf{r}_g - {}^o\mathbf{r}_{g,i}$ is the difference between the ZMP position vector and the FCP position vector, while ${}^o\mathbf{r}_{g,j} - {}^o\mathbf{r}_{g,i}$ is the difference vector between two consecutive FCPs j and i which define a support polygon; $\varepsilon = \pm 1$ is a constant, and n_p is the number of sides of the foot support polygon. The dot product is expressed as $(\cdot)$. In the general case, the foot support polygon is convex-shaped, but in the simplest case, it is a rectangle or a parallelogram. We assume that the foot support polygon is always quadrilateral, which is equivalent to the assumption that $n_p = 4$. A *double support* stance means that both feet touch the ground, while a *single support* stance means that only one foot touches the ground. It is assumed that there are only three cases of foot support polygons (Fig. 4). The first case, the double support stance, describes a large step (wide stride): the heel position of the front leg is in front of the toe position of the back leg (Fig. 4a). The second case, also the double support stance, describes a small step (short stride): the heel position of the front leg is behind the toe position of the back leg (Fig. 4b). The last case, the single support stance, is valid when a single foot touches the ground (Fig. 4c). There can also be variations in the third case, for example, when the foot support polygon covers the area from the heel to the joint between the foot and toes. In this case, the foot support polygon would be spanned by points 1-2-4-3. However, for simplicity, we ignore this configuration of the foot support polygon. Also, you can notice that in the second case we ignore the small region spanned by the inside toe of the front leg and the inside heel

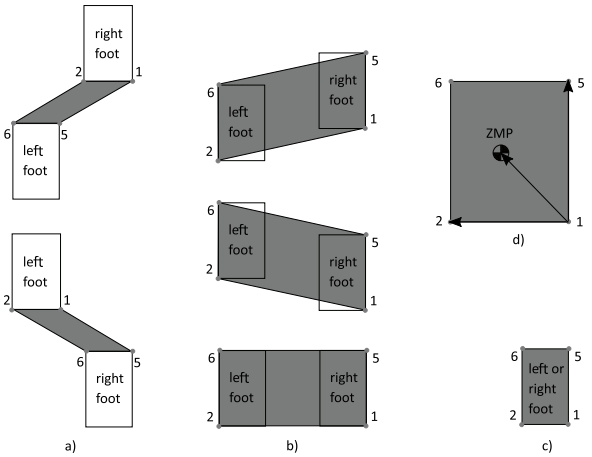


FIGURE 4. Schematic drawing of the foot support polygon (gray area). a) Double support phase: heel position of one foot being in front of the toe position of another foot (top - right foot in front, bottom - left foot in front). b) Double support phase: heel position of one foot being behind the toe position of another foot (top - right foot in front, center - left foot in front, bottom - both feet are on a line). c) Single support phase. d) ZMP should be within the gray area.

of the back leg. The sign ε in (25) depends on whether the dot product in it has a negative or positive value, which in turn depends on the order of FCPs chosen to construct the foot support polygon (Fig. 4d). For example, if in Fig. 4d the order of the FCPs is clockwise 1-2-6-5, then $\varepsilon = -1$, $j = 2$, $i = 1$. If, on the other hand, the order of the FCPs is counterclockwise 1-5-6-2, then $\varepsilon = 1$, $j = 5$, $i = 1$. Alternatively, there are different formulations of the ZMP condition [43].

H. ARM-LEG MOTION COUPLING

This constraint imposes the condition that during walking the arm and leg motions are synchronized so that they swing in anti-phase. For instance, when the right leg moves forward, the right arm moves backward. This happens naturally to preserve the angular momentum of the body along the vertical z_0 axis. The constraint can be written as

$$(\mathbf{r}_{arm} \cdot \hat{m})(\mathbf{r}_{leg} \cdot \hat{m}) \leq 0, \tag{26}$$

where $\mathbf{r}_{arm} \in \mathbb{R}^{3\times 1}$ is the difference between the global positions of the hand and the shoulder. Similarly, $\mathbf{r}_{leg}$ is the difference between the global positions of the toe and the hip. The direction of the human body's forward motion is given as $\hat{m}$. (26) imposes the anti-phase oscillatory motion of the arm and leg on each side of the body during walking. Thus, for a human model, there are two conditions (26): one for the right arm-leg coupling and another for the left arm-leg coupling.

I. KNEE MID-SWING CONSTRAINT

The final constraint is imposed on the knee angle flexion q_{knee} during the mid-swing of the leg. This constraint is simply

$$q_{knee} \geq \frac{\pi}{3}. \tag{27}$$

The aforementioned constraints are imposed on the human body. The initial and final conditions and foot contact constraints are equality constraints in their nature. While the remaining constraints are inequality constraints. However, inequality constraints are more general than equality constraints because any equality constraint $\mathcal{H}(\mathbf{y}) = 0$ from (3) can be converted into two inequality constraints $\mathcal{H}(\mathbf{y}) - \epsilon \leq 0$ and $-\mathcal{H}(\mathbf{y}) - \epsilon \leq 0$. If the tolerance variable $\epsilon = 0$, then we restore the original equality constraint.

VI. SIMULATIONS

To check the proposed approach, we decided to simulate four different motions: still-walk-still, sit-stand, side-step, and jump.

A. STILL-WALK-STILL

In this scenario, the human model makes three steps forward and then stops: the first half-step is with the right leg, the second (third) full step is with the left (right) leg, and the last half-step is with the left leg again. The simulation time is set to $t_{sim} = 3$ s with the discretization sampling time $\Delta t = 0.5$ s and $n_t = 7$. It is assumed that the human model is at rest and at its neutral position at the initial and final times. This corresponds to the initial and the final values of the generalized coordinates and velocities being set to zero, except for the final value of the generalized coordinate corresponding to the global displacement along the forward y_0 axis: $q_2(t_{sim}) = 3L^f_{step}$, where $L^f_{step} = 0.4$ m is the forward step length. The foot contact constraint matrix is specified as $\Lambda = [1\ 1\ 1\ 1\ 0\ 1\ 1] \in \mathbb{R}^{6\times 7}$ for the right foot and $\Lambda = [1\ 1\ 0\ 1\ 1\ 1\ 1] \in \mathbb{R}^{6\times 7}$ for the left foot, where 1 and 0 are the column vectors of the appropriate dimension. All constraints are active. The total number of the optimization variables is 430 and the number of constraints is 2812.

B. SIT-STAND

This scenario corresponds to the human model standing up from the initial sitting position. For this scenario $t_{sim} = 1$ s and $\Delta t = 0.2$ s. It is assumed that the human model is at rest at the initial and final times. This corresponds to the initial values for some of the generalized coordinates being set to the following: $q_1(0) = -0.44$ m, $q_9(0) = -\frac{\pi}{2}$, $q_{10}(0) = \frac{\pi}{2}$, $q_{17}(0) = -\frac{\pi}{2}$, $q_{18}(0) = \frac{\pi}{2}$, $q_{28}(0) = -\frac{\pi}{2}$, $q_{37}(0) = -\frac{\pi}{2}$. The final values for the generalized coordinate are: $q_2(t_{sim}) = 0.44$ m and zero for all others. The initial and final values for all the generalized velocities are zero. All elements of the two foot contact constraint matrices $\Lambda \in \mathbb{R}^{6\times 6}$ are set to 1, because in this motion the feet are always in contact with the ground. The constraints on the pelvis velocity, the arm-leg motion coupling, and the knee mid-swing magnitude are deactivated. In this case, the number of the optimization variables and the number of constraints are 387 and 2508, respectively.

C. SIDE STEP

In this scenario, the human model does a sidestep towards its right side, $t_{sim} = 1$ s and $\Delta t = 0.2$ s. It is also

assumed that the human model is at rest and at its neutral position at the initial and final times. Like the first case, this corresponds to the initial and final values for the generalized coordinates and velocities being set to zero, except for the final value of the generalized coordinate corresponding to the global displacement along the right x_0 axis: $q_3(t_{sim}) = L^s_{step}$, where $L^s_{step} = 0.3$ m is the sidestep length. The foot contact constraint matrices are specified as $\Lambda = [1\,0\,1\,1\,1\,1] \in \mathbb{R}^{6\times6}$ for the right foot and $\Lambda = [1\,1\,1\,1\,0\,1] \in \mathbb{R}^{6\times6}$ for the left foot. For this scenario, the arm-leg motion coupling and the knee mid-swing magnitude constraints are deactivated. In this scenario, the number of optimization variables is 387 and the number of constraints is 2372.

D. JUMP

This scenario describes the human model motion before the jump, with $t_{sim} = 0.6$ s and $\Delta t = 0.2$ s. The objective function was extended by a term describing the jump height of the pelvis point as

$$\tilde{\mathcal{J}}(\mathbf{y}) = \mathcal{J}(\mathbf{y}) - \eta({}^{p}r_{z,g} + \frac{{}^{p}v^2_{z,g}}{2g}), \tag{28}$$

where $\tilde{\mathcal{J}}(\mathbf{y})$ is the modified cost function; $\mathcal{J}(\mathbf{y})$ is from (15); η is the weight coefficient that describes the relative importance of the jump height. We assumed that $\eta = 10^4$ to prioritize the jump height. The z coordinates of the pelvis point position and velocity in the global frame are denoted as ${}^{p}r_{z,g}$ and ${}^{p}v_{z,g}$, respectively. The initial and final values for all the generalized coordinates and velocities are zero. The only exception is the final value of the vertical speed of the pelvis point, which is set to 2 m/s. This value is lower than the measured average take-off speed for vertical jumps [69], [70]. All elements of the two foot contact constraint matrices $\Lambda \in \mathbb{R}^{6\times5}$ are set to 1. The constraints for the pelvis velocity, arm-leg motion coupling, and knee mid-swing magnitude are disabled. The number of optimization variables is 301 and the number of constraints is 1816.

The knot vector $[0\,0\,0\,0\,t_1 \ldots t_{sim}\ t_{sim}\ t_{sim}\ t_{sim}]$, where the multiplicity of the first and the last element is 4, is used for the discretization using the trigonometric splines for all cases. This is necessary to have nonzero values for the generalized coordinates, velocities, and accelerations at the initial and final times. The boundary conditions (18) on the generalized coordinates are fixed for all cases as follows: $q_{l,1:3} = [-0.05\ 0\ -0.05]$ m (lower) and $q_{u,1:3} = [0.05\ 2\ 0.05]$ m (upper) constraints on the global pelvis displacement, $q_{l,10} = q_{l,18} = 0$ (lower) and $q_{u,28} = q_{u,37} = 0$ (upper) constraint on the knee bending, $q_{l,28} = q_{l,37} = -\frac{2\pi}{3}$ lower constraint on the elbow bending. These limits are adjusted accordingly for the sidestep and sit-stand cases. All other lower and upper joint limits are set to $\pm\frac{\pi}{3}$ rad. All lower and upper bounds on the generalized velocities are set to ±5 rad/s, respectively. The limits on all torques (19) are set to $\mathbf{Q}_l = -\mathbf{Q}_u = -200$ Nm, except for the bound on the global force along the z_0 axis, which is set to $1.4m_{total}g$. Here $m_{total} = 77.5$ kg is the total mass of the body and $g = 9.81$ m/s^2. The numeric values of physical parameters (link lengths and masses) of the model were taken from [7]. We intentionally do not provide these values here, so that our description does not depend on a specific model. The torque constraints are rough limits, and obviously most human joints cannot generate this amount of torque. Imposing real torque limits would of course lead to more realistic movements. Additionally, imposing state-dependent torque limits was shown to provide better solutions [71]. However, our goal is to have a valid solution with minimum input. Therefore, the torque limits are not set to reflect the physical limitations of joints, but to facilitate the search for a local minimum in the NLP. In addition, the torques in the obtained solutions are on the order of 5-10 Nm or less for most of the joints, which is within reasonable range. The NLP was solved using `fmincon` solver of MATLAB, which is considered rather slow compared to other NLP solvers. However, the main advantages of using MATLAB are related to its widespread use and the simplicity of its programming language. Gradients of both the objective function and the constraint functions are specified. Next, we briefly describe some practical tips. The `interior-point`, `sqp`, and `active-set` algorithms were used for the NLP. Our approach is to first try the `interior-point` algorithm, if that fails, switch to `sqp`, and finally, if that also fails, try `active-set`. Also, it might be helpful to activate the constraints gradually. For example, one could activate only a subset of the constraints on the first run and then use the obtained feasible solution as the initial condition for the next run, where more constraints are activated. It was found that running two NLPs consecutively in this manner sometimes takes less time than running a single NLP with all constraints active. To obtain the initial feasible solution, it might be better to impose a constant or zero objective function first. With the constant objective function, the NLP tries to find any feasible solution. After that, the obtained feasible solution can be used as an initial condition with a true objective function. Constraint tolerance and step tolerance settings were set to $4 \cdot 10^{-2}$ and 10^{-9}, respectively. The simulations were performed on a ThinkPad notebook with an Intel Core i5-10310U CPU processor and 16 GB RAM. The NLP problem solution results are given below.

VII. EXPERIMENTS

Experiments were performed to confirm the validity of the proposed model. To this end, we track human motion with an optical motion capture (MOCAP) system. We use the Qualisys MOCAP system that works with six Miqus M3 MOCAP cameras and the Qualisys Track Manager (QTM) software. Miqus M3 cameras are dual-mode cameras that use two MP sensors (340 records per second) to capture fast-moving objects with high accuracy [72]. The Qualisys MOCAP system allowed us to record the position, rotation, and pose of rigid bodies with 6 DOF. In our measurement campaign, we recorded human walking activity

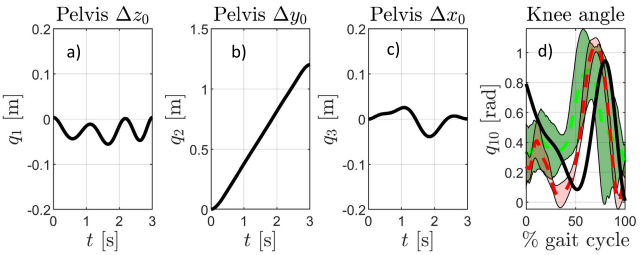


FIGURE 5. Trajectories for: a) vertical, b) forward horizontal, c) lateral displacements of the pelvis point in the global reference frame, d) knee joint angle as a function of the gait cycle. (black curve: simulation, red curve: from literature, green curve: experiment). Still-walk-still motion.

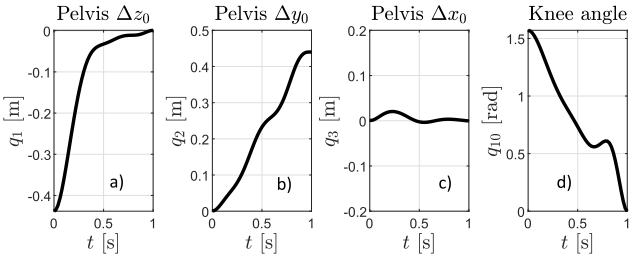


FIGURE 6. Trajectories for: a) vertical, b) forward horizontal, c) lateral displacements of the pelvis point in the global reference frame, d) knee joint angle. Sit-stand motion.

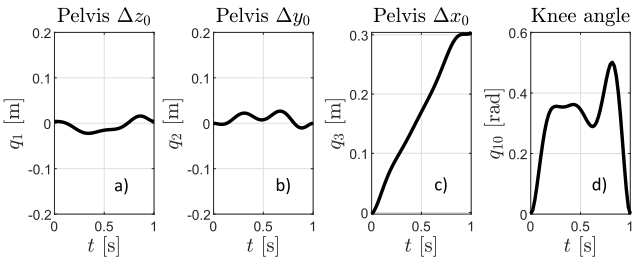


FIGURE 7. Trajectories for: a) vertical, b) forward horizontal, c) lateral displacements of the pelvis point in the global reference frame, d) knee joint angle. Side step motion.

with automatic identification markers (AIM) to identify and label the trajectories of human body segments. As an AIM model, we implemented a Qualisys animation marker set with 42 reflecting markers, where a 38 years old healthy male participant with a height of 1.75 m and a weight of 90 kg performed five trials of four-step level walks. Each reflecting marker was placed on the actor's body segments according to the animation marker set guide. Our experiments with the human walk scenario were conducted in accordance with accepted ethical standards and guidelines for the use of research participants to protect their rights and welfare.

VIII. RESULTS

The NLP for the still-walk-still motion was solved in three stages. In the first stage, the NLP with the zero objective function and constraints (17)-(21) was solved in 85 seconds using the `active-set` algorithm and the initial conditions

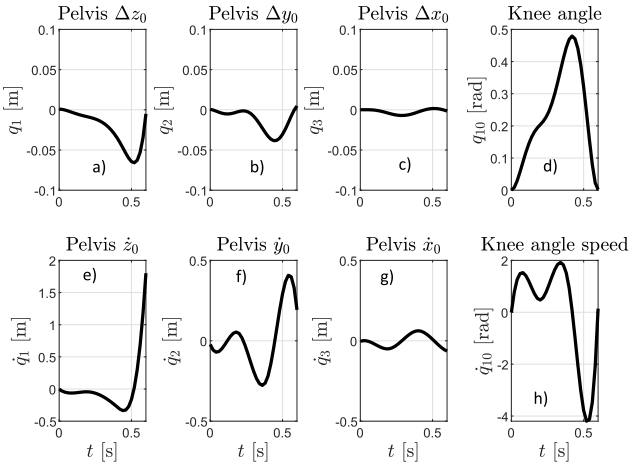


FIGURE 8. Trajectories for displacements and speeds of the pelvis point in vertical (a and e), forward horizontal (b and f), and lateral directions (c and g) in the global reference frame. Knee angles and velocity (d and h). Jump motion.

given in Sec. VI-A. At this stage, the attempt to solve the NLP with the `interior-point` and `sqp` algorithms did not produce the desired results. In the second stage, the objective function (15) and the remaining constraints were enforced, and the resulting NLP was solved using the `interior-point` in 15 hours and 39 minutes. In the final stage, the NLP was solved using the `sqp` algorithm in 3 hours and 22 minutes. In the second and third stages, the initial conditions for the NLP were taken from the corresponding solutions of the previous stage. Even though the time to solve the NLP is several hours, it is compatible with the reported solution times. For instance, an OCP for a lifting motion of a simpler 17 DOF two-dimensional human model was solved in 4 to 8 hours [73]. The snapshots of the obtained NLP solution are shown in Fig. 9 for the walking motion. The human body qualitatively performs the intended motion. The leg motions correspond to the imposed foot contact conditions. There are also discrepancies with real human motion. For example, the human body tilts both arms backwards. This could be explained by the low numerical values of the arm-leg motion coupling constraints. We employed the same weights for all constraints. Better results would be achieved if weighted constraints were used instead (high weighting for important constraints and vice versa). Different solutions were identified to test the dependence of the time required to solve the NLP on the imposed constraints and the objective function. However, no clear pattern emerged in finding the correct solution. This could be due to the highly nonlinear nature of the objective function, where the performance of the NLP depends not only on the initial condition, but also on the solution path (how and in what order constraints and objective functions are applied). The shifts of the pelvis point with respect to its initial position in the global reference frame are shown in Fig. 5. The forward y_0 displacement of the pelvis point is uniform, while the lateral x_0 and vertical z_0 displacements are negligible. Also shown is the knee

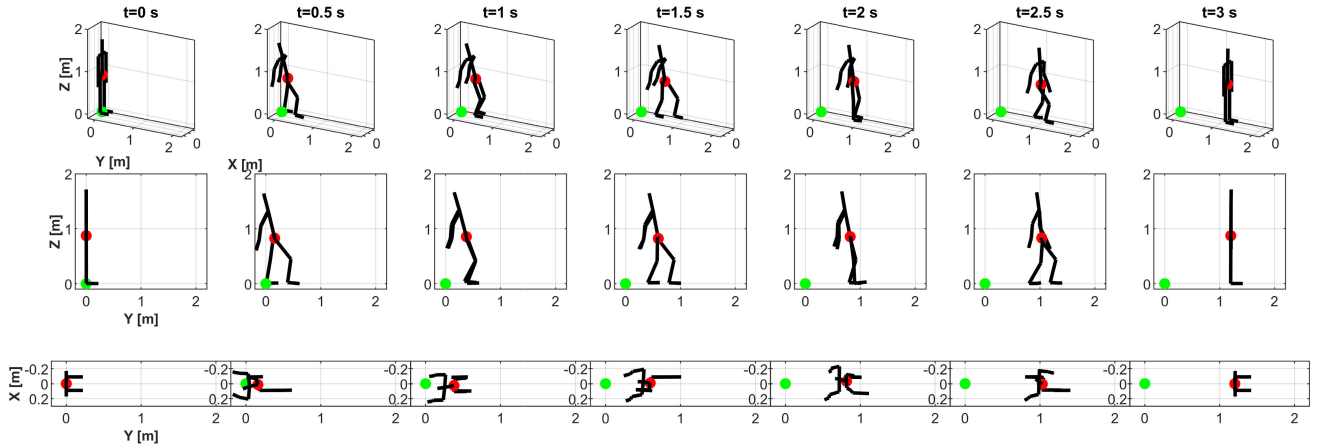


FIGURE 9. Frames of the walking movement: Top row: global view, middle row: sagittal plane view, bottom row: transverse plane view. (Origin of the global reference frame: green dot, pelvic point: red dot).

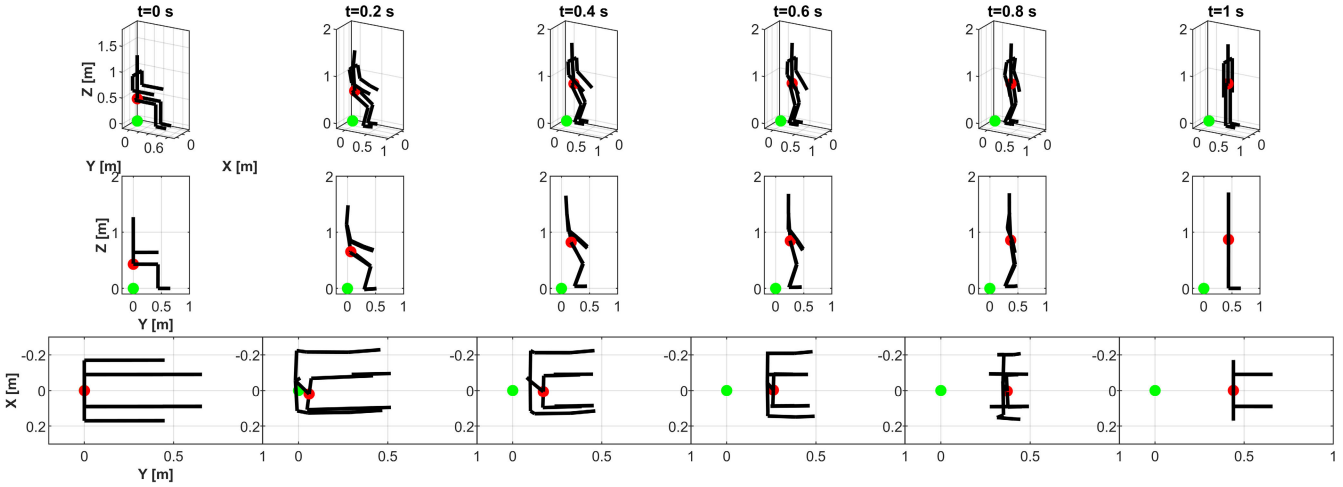


FIGURE 10. Frames of the sit-stand movement: Top row: global view, middle row: sagittal plane view, bottom row: transverse plane view. (Origin of the global reference frame: green dot, pelvic point: red dot).

angle as a function of the gait (Fig. 5d). The black solid line shows the simulation results for the right knee. The red dotted line and red area are the average and standard deviation of the measured right knee angle during level walking of Subject 6, obtained from open-source data [74]. The data has 11 trials. The green dotted line and green areas are the average and standard deviation of the knee angle measured in our lab. It can be observed that the simulated knee angle qualitatively matches the experimental knee angle trajectories, indicating that further simulations are valid. The discrepancy between the simulation and experiments can be attributed to differences between the human model and real subjects, simple foot-ground interaction constraints, and the specified initial and final conditions. For example, the simulated knee angle is not periodic because the motion consists of only a single stride (two steps) with the right leg and then the human model comes to rest due to the imposed final condition. Additionally, the dynamic analysis is very sensitive to the values of the human segment parameters [75]. There are also discrepancies between the two experiments due to different human subjects.

The results for the sit-stand motion are shown as snapshots in Fig. 10. The NLP was solved in 9 minutes using the `interior-point` algorithm in `fmincon`. A difference from the real human motion can be seen in the frames $t = 0.2, 0.4$ s, in which the human model leans slightly backward during the stand-up motion instead of leaning forward. The motion viewed in the sagittal plane is symmetric. The z, y, and x components of the displacement of the pelvis point with respect to its initial position in the global reference frame are shown in Fig. 6. As expected, both the vertical (z_0) and the forward horizontal (y_0) displacements of the pelvis are equal to the distance between the knee joint and the hip joint. This shows that during stand-up movement, the vertical displacement of the pelvis point is compensated by its horizontal displacement. The lateral (x_0) displacement of

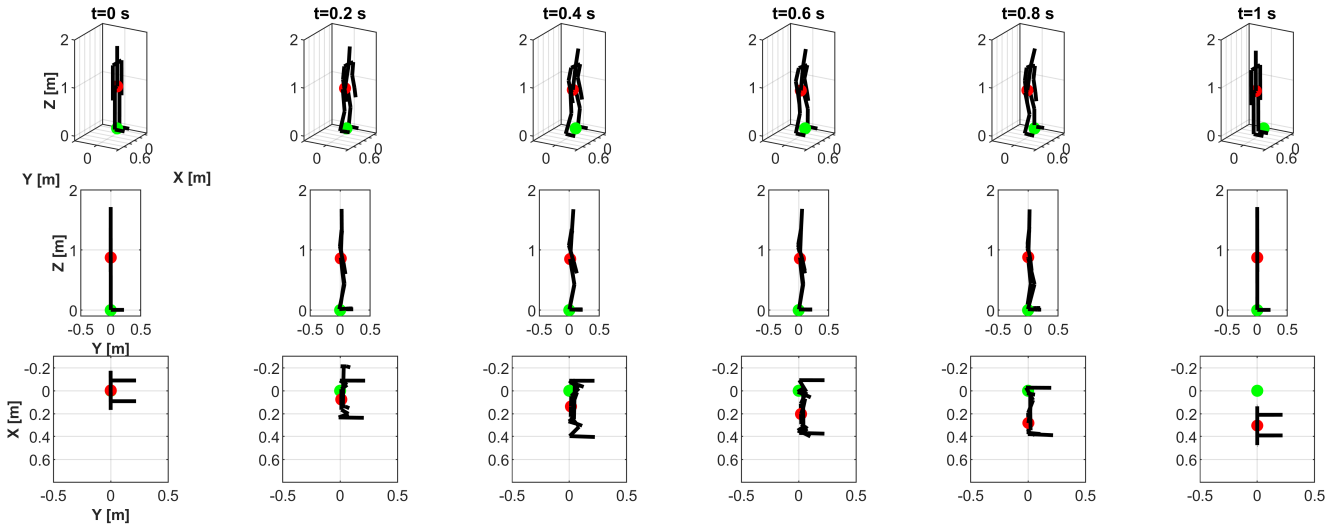


FIGURE 11. Frames of the side step movement: Top row: global view, middle row: sagittal plane view, bottom row: transverse plane view. (Origin of the global reference frame: green dot, pelvic point: red dot).

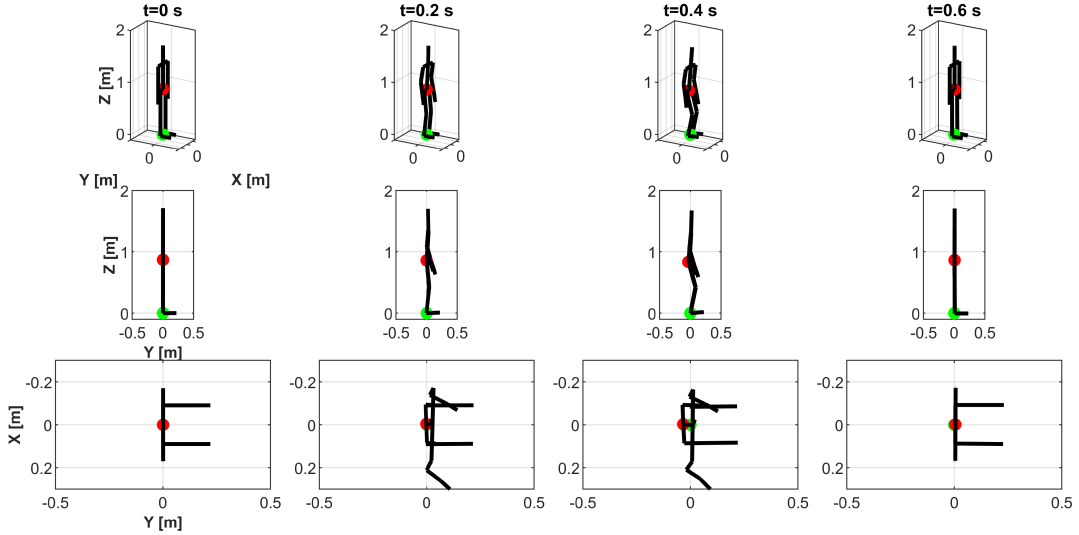


FIGURE 12. Frames of the jump movement: Top row: global view, middle row: sagittal plane view, bottom row: transverse plane view. (Origin of the global reference frame: green dot, pelvic point: red dot).

the pelvis point is negligible. The right knee joint angle is also shown in Fig. 6d. The knee angle decreases from an initial 90° to 0°. However, the discrepancy with the experimental knee joint angles is also observed. The measured knee joint angle gradually decreases during the stand-up movement [76], whereas in our simulation there is a small increase at the end of the movement, Fig. 6d.

The results for the sidestep motion are shown in Fig. 11. The NLP was solved in an impressive 9 seconds using the `interior-point` algorithm. Here, the motion looks qualitatively correct: during the sidestep, the knees bend, the elbows also bend upward, and the head tilts slightly forward. The three components of the vector of the pelvis point displacement and the knee joint angle are shown in Fig. 7. As expected, both the vertical and horizontal forward displacement of the pelvis point are negligible. The maximum vertical displacement is 1.6 cm, which is within the range of the experimental vertical displacement of the pelvis center of gravity [77]. The knee angle flexion does not vary as much as during the standing-up and walking movements. The experimental knee flexion angle during sidestep (also called side cut) motion is a concave function (when plotted as a function of a cut cycle) as reported in [78]. In our simulation, the cut cycle starts at 0.4 s and ends at 1 s. Therefore, the simulated knee angle qualitatively matches the experimental trajectory starting only from 0.6 s, Fig. 7d).

The lateral x_0 displacement is uniform and corresponds to the given sidestep length.

The results for the jump motion are shown in Fig. 12. The human model sits correctly before jumping vertically upwards. The trajectories of the pelvis point and the right knee angle are shown together with their corresponding velocities in Fig. 8. The obtained knee joint flexion is in qualitative agreement with the experimental knee joint trajectory during the jump motion [79]. We also point out that the knee flexion phase is longer than the knee extension phase (see Fig. 8d), which is also confirmed by the experiments reported in [79]. The vertical jump speed is 1.8 m/s, which results in a jump height of 16 cm for the pelvis point. In this case, the NLP was solved in 1 hour and 25 minutes using the `interior-point` algorithm.

One limitation of the proposed method is that it does not calculate the true joint torques. To do that, the generalized forces corresponding to the global DOF should be used to recalculate the joint torques as in [7]. However, this would require additional complex computations without any significant impact on the full-body human dynamics. First, these additional calculations would essentially distribute the global forces only to weight-bearing joints (to joints in weight-bearing kinematic branches, e.g., the legs). Second, accurate joint torque estimates are necessary for rehabilitation medicine and other related fields, but our project is concerned with the determination of dynamically valid motions. However, it is straightforward to extend the proposed method to output the correct joint torques. The simple objective function and constraints used in the NLP may be a cause of discrepancy between obtained simulation results and experiments. More realistic motions can be synthesized if multiple objectives are combined and/or more constraints are imposed, as in [12] and [50]. Another reason for the discrepancy could be the simple human skeletal model. Considering that using a more advanced human musculoskeletal model with a linearized Hill muscle model leads to unrealistic results [80], our results look promising. Achieving valid and realistic human motions is a trade-off between computational power, manual effort, and the quality of the solution. The proposed human motion synthesis method uses simple objective functions and constraints, an elementary foot-ground interaction model, and the simplest human skeletal model, thus trading off solution quality for computational efficiency and labor. This partially explains the discrepancy between the simulation results obtained and the experiments. It is hoped that the work will help advance computational human dynamics by inspiring others to initiate research in this area.

IX. CONCLUSION

The human motion synthesis framework, which can be easily implemented in a user-friendly MATLAB environment, is proposed in this work. We use MATLAB because it is easy for debugging and many engineering students are familiar with it. In contrast to the commonly used B-splines, the generalized coordinates and velocities are discretized with trigonometric splines. Human motion synthesis is achieved through the optimization method. Four simulations of the human motion were synthesized: walking, sit-stand, sidestep, and jump. Level walking experiments were performed to validate the obtained walking simulation results. The obtained results and the NLP solution time were presented and discussed. The results indicate that trigonometric splines can be used to synthesize human motion. However, the synthesis of human motion is still a challenging problem, and further work is needed to refine and improve its quality. Judging by the recent trends, human motion synthesis now is mostly achieved through deep learning, reinforcement learning, or other machine learning techniques. It might be that breakthroughs in human motion synthesis happen through these techniques. Nonetheless, other methods to achieve human motion synthesis should still be explored. The final goal of human motion synthesis is to bring to the level that even non-skilled users can generate valid and physically plausible motions with less effort. We hope that in the future computational human dynamics will play a similar role as computational fluid dynamics or computational electrodynamics.

A promising extension of this work could be the development of interactive motion synthesis from basic motion primitives or templates. Another direction for future work is to develop a framework for automatically estimating human model parameters based on experimental motion data. Automatic adjustment of lower and upper boundaries of the constraints to generate more realistic human motion is another area for future research.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interests regarding the publication of this article.

REFERENCES

[1] M. Ikawa, E. Ueda, A. Yuguchi, G. A. G. Ricardez, M. Ding, J. Takamatsu, and T. Ogasawara, "Quantification of elegant motions for receptionist Android robot," in *Proc. Int. Conf. Digital Hum. Modeling. Applied Health, Safety, Ergonom., Risk Management, Ergonom. Design*, 2017, pp. 435–446.

[2] T. Liu, Y. Inoue, K. Shibata, and K. Shiojima, "A mobile force plate and three-dimensional motion analysis system for three-dimensional gait assessment," *IEEE Sensors J.*, vol. 12, no. 5, pp. 1461–1467, May 2012.

[3] J. H. Jung and J. F. Veneman, "Real time computation of centroidal momentum while human walking in the lower limbs rehabilitation exoskeleton: Preliminary trials," in *Proc. IEEE 16th Int. Conf. Rehabil. Robot. (ICORR)*, Jun. 2019, pp. 721–726.

[4] S. Mihcin, H. Kose, S. Cizmeciogullari, S. Ciklacandir, M. Kocak, A. Tosun, and A. Akan, "Investigation of wearable motion capture system towards biomechanical modelling," in *Proc. IEEE Int. Symp. Med. Meas. Appl. (MeMeA)*, Jun. 2019, pp. 1–5.

[5] A. R. Anwary, H. Yu, and M. Vassallo, "Optimal foot location for placing wearable IMU sensors and automatic feature extraction for gait analysis," *IEEE Sensors J.*, vol. 18, no. 6, pp. 2555–2567, Mar. 2018.

[6] T. X. Tran, C.-K. Kang, and S. L. Mathis, "Lower-gait tracking mobile application: A case study of lower body motion capture comparison between vicon T40 system and apple augmented reality," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Dec. 2020, pp. 2654–2656.

[7] J. S. A. Karim and A. Abdel-Malek, *Human Motion Simulation: Predictive Dynamics*, 1st ed. Amsterdam, The Netherlands: Elsevier, 2013.

[8] J. Rasmussen, "Challenges in human body mechanics simulation," *Proc. IUTAM*, vol. 2, pp. 176–185, Jan. 2011.

[9] N. Avazov, R. Hicheri, and M. Pätzold, "A trajectory-driven SISO mm-wave channel model for a human activity recognition," in *Proc. 17th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2021, pp. 133–138.

[10] B. R. Umberger and R. H. Miller, *Optimal Control Modeling of Human Movement*. Berlin, Germany: Springer, 2017, pp. 1–22.

[11] I. Mordatch, M. de Lasa, and A. Hertzmann, "Robust physics-based locomotion using low-dimensional planning," in *Proc. ACM SIGGRAPH Papers (SIGGRAPH)*, 2010, pp. 1–8.

[12] M. A. Borno, M. D. Lasa, and A. Hertzmann, "Trajectory optimization for full-body movements with complex contacts," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 8, pp. 1405–1414, Aug. 2013.

[13] M. Ezati, B. Ghannadi, and J. McPhee, "A review of simulation methods for human movement dynamics with emphasis on gait," *Multibody Syst. Dyn.*, vol. 47, no. 3, pp. 265–292, Nov. 2019.

[14] C. de Boor, *A Practical Guide to Splines*. New York, NY, USA: Springer, 1978.

[15] S. Biswas and B. C. Lovell, *B-Splines and Its Applications*. London, U.K.: Springer, 2008.

[16] Q. Wang, Y.-J. Xiang, J. Arora, and K. Abdel-Malek, "Alternative formulations for optimization-based human gait planning," in *Proc. 48th AIAA/ASME/ASCE/AHS/ASC Struct., Struct. Dyn., Mater. Conf.*, 2012, pp. 1–12.

[17] A. C. Fang and N. S. Pollard, "Efficient synthesis of physically valid human motion," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 417–426, Jul. 2003.

[18] D. Holden, J. Saito, and T. Komura, "A deep learning framework for character motion synthesis and editing," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–11, Jul. 2016.

[19] X. B. Peng, G. Berseth, K. Yin, and M. V. D. Panne, "Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning," *ACM Trans. Graph.*, vol. 36, no. 4, p. 41, Jun. 2017.

[20] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "DeepMimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–14, Jul. 2018.

[21] C. K. Liu and Z. Popović, "Synthesis of complex dynamic character motion from simple animations," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 408–416, Jul. 2002.

[22] Y. Ye and C. K. Liu, "Optimal feedback control for character animation using an abstract model," in *Proc. ACM SIGGRAPH Papers (SIGGRAPH)*, 2010, pp. 1–9.

[23] K. Yin, K. Loken, and M. Van de Panne, "Simbicon: Simple biped locomotion control," *ACM Trans. Graph.*, vol. 26, p. 105, Jul. 2007.

[24] P. Faloutsos, M. van de Panne, and D. Terzopoulos, "Composable controllers for physics-based character animation," in *Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn.*, Aug. 2001, pp. 251–260.

[25] W. Yu, G. Turk, and C. K. Liu, "Learning symmetric and low-energy locomotion," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–12, Jul. 2018.

[26] A. Ghosh, N. Cheema, C. Oguz, C. Theobalt, and P. Slusallek, "Synthesis of compositional animations from textual descriptions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 1376–1386.

[27] J. Ondras, O. Celiktutan, P. Bremner, and H. Gunes, "Audio-driven robot upper-body motion synthesis," *IEEE Trans. Cybern.*, vol. 51, no. 11, pp. 5445–5454, Nov. 2021.

[28] S. Agrawal and M. van de Panne, "Task-based locomotion," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–11, Jul. 2016.

[29] K. Xie, T. Wang, U. Iqbal, Y. Guo, S. Fidler, and F. Shkurti, "Physics-based human motion estimation and synthesis from videos," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 11512–11521.

[30] L. Liu, J. L. Cooper, and D. H. Ballard, "Computational modeling: Human dynamic model," *Frontiers Neurorobotics*, vol. 15, pp. 1–16, Sep. 2021.

[31] J. Won, D. Gopinath, and J. Hodgins, "Control strategies for physically simulated characters performing two-player competitive sports," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 1–11, Jul. 2021.

[32] S. Liu, J. Merel, Z. Wang, S. M. A. Eslami, D. Hennes, W. M. Czarnecki, Y. Tassa, S. Omidshafiei, A. Abdolmaleki, N. Y. Siegel, and L. Hasenclever, "From motor control to team play in simulated humanoid football," *Sci. Robot.*, vol. 7, no. 69, Aug. 2022, Art. no. eabo0235.

[33] B. Haworth, G. Berseth, S. Moon, P. Faloutsos, and M. Kapadia, "Deep integration of physical humanoid control and crowd navigation," in *Proc. Motion, Interact. Games*, Oct. 2020, pp. 1–10.

[34] M. Althoff, M. Mayer, and R. Müller, "Automatic synthesis of human motion from temporal logic specifications," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 4040–4046.

[35] Z. Xie, H. Y. Ling, N. H. Kim, and M. Panne, "ALLSTEPS: Curriculum-driven learning of stepping stone skills," *Comput. Graph. Forum*, vol. 39, no. 8, pp. 213–224, Dec. 2020.

[36] C. K. Chow and D. H. Jacobson, "Studies of human locomotion via optimal programming," *Math. Biosciences*, vol. 10, nos. 3–4, pp. 239–306, Apr. 1971.

[37] D. G. Hull, "Conversion of optimal control problems into parameter optimization problems," *J. Guid. Control Dyn.*, vol. 20, no. 1, pp. 57–60, Jan. 1997.

[38] A. V. Rao, "A survey of numerical methods for optimal control," *Adv. Astron. Sci.*, vol. 135, no. 1, pp. 1–32, Jan. 2010.

[39] Y. Xiang, J. S. Arora, and K. Abdel-Malek, "Physics-based modeling and simulation of human walking: A review of optimization-based and other approaches," *Struct. Multidisciplinary Optim.*, vol. 42, pp. 1–23, Jul. 2010.

[40] R. Boulic, N. M. Thalmann, and D. Thalmann, "A global human walking model with real-time kinematic personification," *Vis. Comput.*, vol. 6, no. 6, pp. 344–358, Nov. 1990.

[41] F. C. Anderson and M. G. Pandy, "Dynamic optimization of human walking," *J. Biomech. Eng.*, vol. 123, no. 5, pp. 381–390, Oct. 2001.

[42] C. K. Liu, A. Hertzmann, and Z. Popović, "Learning physics-based motion style with nonlinear inverse optimization," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 1071–1081, Jul. 2005.

[43] G. Bessonnet, J. Marot, P. Seguin, and P. Sardain, "Parametric-based dynamic synthesis of 3D-gait," *Robotica*, vol. 28, no. 4, pp. 563–581, Jul. 2010.

[44] Y. Xiang, J. S. Arora, S. Rahmatalla, and K. Abdel-Malek, "Optimization-based dynamic human walking prediction: One step formulation," *Int. J. Numer. Methods Eng.*, vol. 79, no. 6, pp. 667–695, Feb. 2009.

[45] M. Ackermann and A. J. van den Bogert, "Optimality principles for model-based prediction of human gait," *J. Biomech.*, vol. 43, no. 6, pp. 1055–1060, 2010.

[46] M. G. Pandy, F. C. Anderson, and D. G. Hull, "A parameter optimization approach for the optimal control of large-scale musculoskeletal systems," *J. Biomechanical Eng.*, vol. 114, no. 4, pp. 450–460, Nov. 1992.

[47] F. C. Anderson and M. G. Pandy, "A dynamic optimization solution for vertical jumping in three dimensions," *Comput. Methods Biomech. Biomed. Eng.*, vol. 2, no. 3, pp. 201–231, Jan. 1999.

[48] S. Porsa, Y.-C. Lin, and M. G. Pandy, "Direct methods for predicting movement biomechanics based upon optimal control theory with implementation in OpenSim," *Ann. Biomed. Eng.*, vol. 44, no. 8, pp. 2542–2557, Aug. 2016.

[49] M. L. Kaplan and J. H. Heegaard, "Predictive algorithms for neuromuscular control of human locomotion," *J. Biomechanics*, vol. 34, no. 8, pp. 1077–1083, Aug. 2001.

[50] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 1–8, Jul. 2012.

[51] T. Geijtenbeek, M. van de Panne, and A. F. van der Stappen, "Flexible muscle-based locomotion for bipedal creatures," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 1–11, Nov. 2013.

[52] S. Jain, Y. Ye, and C. K. Liu, "Optimization-based interactive motion synthesis," *ACM Trans. Graph.*, vol. 28, no. 1, pp. 1–12, Feb. 2009.

[53] S. Coros, P. Beaudoin, and M. van de Panne, "Generalized biped walking control," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 1–9, Jul. 2010.

[54] I. Mordatch, M. de Lasa, and A. Hertzmann, "Robust physics-based locomotion using low-dimensional planning," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 1–8, Jul. 2010.

[55] H. Geyer and H. Herr, "A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 18, no. 3, pp. 263–273, Jun. 2010.

[56] J. M. Wang, S. R. Hamner, S. L. Delp, and V. Koltun, "Optimizing locomotion controllers using biologically-based actuators and objectives," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 1–11, Jul. 2012.

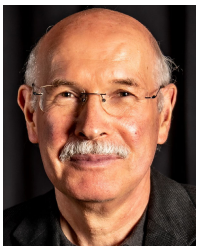
[57] Y. Yang, V. Ivan, W. Merkt, and S. Vijayakumar, "Scaling sampling-based motion planning to humanoid robots," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2016, pp. 1448–1454.

[58] N. Heess, T. B. Dhruva, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. M. Ali Eslami, M. Riedmiller, and D. Silver, "Emergence of locomotion behaviours in rich environments," 2017, *arXiv:1707.02286*.
[59] D. Morante, M. S. Rivo, and M. Soler, "A survey on low-thrust trajectory optimization approaches," *Aerospace*, vol. 8, no. 3, p. 88, Mar. 2021.
[60] E. Bertolazzi, F. Biral, and M. Da Lio, "Symbolic-numeric efficient solution of optimal control problems for multibody systems," *J. Comput. Appl. Math.*, vol. 185, no. 2, pp. 404–421, Jan. 2006.
[61] O. von Stryk and R. Bulirsch, "Direct and indirect methods for trajectory optimization," *Ann. Oper. Res.*, vol. 37, no. 1, pp. 357–373, 1992.
[62] B. A. Conway, "A survey of methods available for the numerical optimization of continuous dynamic systems," *J. Optimiz. Theory App.*, vol. 152, no. 2, pp. 271–306, Feb. 2012.
[63] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Xia, "A review of optimization techniques in spacecraft flight trajectory design," *Prog. Aerosp. Sci.*, vol. 109, Aug. 2019, Art. no. 100543.
[64] J. T. Betts, *Practical Methods for Optimal Control Using Nonlinear Programming*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2001.
[65] I. J. Schoenberg, "On trigonometric spline interpolation," *J. Math. Mech.*, vol. 13, no. 5, pp. 795–825, 1964.
[66] T. Lyche and R. Winther, "A stable recurrence relation for trigonometric B-splines," *J. Approximation Theory*, vol. 25, no. 3, pp. 266–279, Mar. 1979.
[67] G. Walz, "Identities for trigonometric B-splines with an application to curve design," *BIT Numer. Math.*, vol. 37, no. 1, pp. 189–201, Mar. 1997.
[68] L. Liu and D. Ballard, "Humans use minimum cost movements in a whole-body task," *Sci. Rep.*, vol. 11, no. 1, pp. 2045–2322, Oct. 2021.
[69] P. Luhtanen and P. V. Komi, "Segmental contribution to forces in vertical jump," *Eur. J. Appl. Physiol. Occupational Physiol.*, vol. 38, no. 3, pp. 181–188, Sep. 1978.
[70] J. Conlon, G. G. Haff, S. Nimphius, T. Tran, and R. Newton, "Vertical jump velocity as a determinant of speed and agility performance," *J. Austral. Strength Conditioning*, vol. 21, no. S2, pp. 88–90, 2013.
[71] Y. Jiang, T. Van Wouwe, F. De Groote, and C. K. Liu, "Synthesis of biologically realistic human motion using joint torque actuation," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–12, Jul. 2019.
[72] *Qualisys*. Accessed: Sep. 16, 2022. [Online]. Available: https://www.qualisys.com
[73] P. Manns, M. Sreenivasa, M. Millard, and K. Mombaur, "Motion optimization and parameter identification for a human and lower back exoskeleton model," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1564–1570, Jul. 2017.
[74] T. Lencioni, I. Carpinella, M. Rabuffetti, A. Marzegan, and M. Ferrarin, "Human kinematic, kinetic and EMG data during different walking and stair ascending and descending tasks," *Sci. Data*, vol. 6, no. 1, p. 309, Dec. 2019.
[75] A. L. Sheets, S. Corazza, and T. P. Andriacchi, "An automated image-based method of 3D subject-specific body segment parameter estimation for kinetic analyses of rapid movements," *J. Biomechanical Eng.*, vol. 132, no. 1, Dec. 2009, Art. no. 011004.
[76] D. Tarnita, A. Petcu, V. Ontica, D. Prunoiu Diana, and D. N. Tarnita, "Experimental study of sit-to-stand kinematics in healthy, osteoarthritic and prosthetic knee," *IOP Conf. Mater. Sci. Eng.*, vol. 997, no. 1, Dec. 2020, Art. no. 012092.
[77] N. D. Neckel, N. Blonien, D. Nichols, and J. Hidler, "Abnormal joint torque patterns exhibited by chronic stroke subjects while walking with a prescribed physiological gait pattern," *J. Neuroeng. Rehabil.*, vol. 5, no. 1, p. 19, Sep. 2008.
[78] S. M. Sigward and C. M. Powers, "The influence of gender on knee kinematics, kinetics and muscle activation patterns during side-step cutting," *Clin. Biomechanics*, vol. 21, no. 1, pp. 41–48, Jan. 2006.
[79] D. I. Caruntu and R. Moreno, "Human knee inverse dynamics model of vertical jump exercise," *J. Comput. Nonlinear Dyn.*, vol. 14, no. 10, Oct. 2019, Art. no. 101005.
[80] H. Hatze and P. A. Huijing, *Progression of Musculoskeletal Models Toward Large-Scale Cybernetic Myoskeletal Models*. New York, NY, USA: Springer, 2000, pp. 425–439.

ALTAY ZHAKATAYEV (Member, IEEE) received the Ph.D. degree from Nazarbayev University, Astana, Kazakhstan, in 2020. Previously, he worked as a Research Assistant at the Advanced Robotics and Mechatronics Systems (ARMS) Laboratory, Nazarbayev University. Since November 2020, he has been working as a Postdoctoral Researcher with the ICT Department, University of Agder, Kristiansand, Norway. His research interests include optimal control, analytical mechanics, general physics, and mathematics.

NURILLA AVAZOV (Member, IEEE) received the B.Sc. (Hons.) and M.Sc. degrees in telecommunications from the Tashkent University of Information Technologies, Uzbekistan, in 2005 and 2007, respectively, and the Ph.D. degree in telecommunications engineering and computer science from the University of Agder, Grimstad, Norway, and the University of Auckland, in 2015 and 2021, respectively. From 2017 to 2020, he was a Research Fellow at the School of Computer Science, University of Auckland. He is currently working as a Researcher with the University of Agder. He has authored and coauthored numerous technical papers in various journals and conferences. His research interests include wireless communications, channel modeling, machine learning, time series analysis, the Internet of Things, human activity recognition, and eHealth applications.

YURIY ROGOVCHENKO received the M.S. degree (Hons.) from Kyiv State University, in 1983, and the Ph.D. degree in differential equations from the Institute of Mathematics, Kyiv, in 1987. He was a Regular Associate at Abdus Salam ICTP, Trieste, Italy, from 2004 to 2011. He received CNR Research Scholarship (1993 and 1995) and CNR-NATO Guest Fellowship (1997). He also received the Sørlandet Kompetansefonds Research Award, in 2016. He is a referee for over 70 international journals. He was Coordinator of PLATINUM (Partnership for Learning and Teaching in University Mathematics) (Erasmus+ Strategic Partnership in Higher Education). He has NFR Project Cooperative Human Activity Recognition and Localization for Healthcare and Well-being. He served as an Editor for 11 international journals in mathematics.



MATTHIAS PÄTZOLD (Senior Member, IEEE) received the Dipl.Ing. and Dr.Ing. degrees in electrical engineering from Ruhr University Bochum, Bochum, Germany, in 1985 and 1989, respectively, and the Habilitation degree in communications engineering from the Hamburg University of Technology, Hamburg, Germany, in 1998. From 1990 to 1992, he was at ANT Nachrichtentechnik GmbH, Backnang, Germany, where he was involved in digital satellite communications. From 1992 to 2001, he was at the Department of Digital Networks, Hamburg University of Technology. Since 2001, he has been a Full Professor of mobile communications with the University of Agder, Norway. He has authored several books and numerous technical articles. His publications received 14 Best Paper Awards.