Humanoid Motion Optimization

Katja Mombaur

Contents
1 Introduction ............................................................... 2
2 Related Research: Humanoid Optimization Based on Template Models ............... 6
3 Modeling Humanoid Robots as Multibody Systems ........................................ 7
4 Humanoid Motion Optimization by Solution of Optimal Control Problems ............. 11
  4.1 Formulation of Optimal Control Problems for Motion Synthesis and
      Motion Reconstruction ............................................. 12
  4.2 Objective Functions for Humanoid Motion Synthesis ................................. 13
  4.3 Objective Function for Human Motion Imitation by Humanoids ..................... 14
  4.4 Constraints to be Considered in the Optimal Control Problem ...................... 16
  4.5 Solution of Optimal Control Problems ......................................... 17
5 Examples of Successful Applications of Optimization for Whole-Body
      Motion of Humanoids and Other Anthropomorphic Systems ....................... 19
6 Combining Optimization with Learning and Movement Primitives for Motion
      Optimization ................................................................ 24
  6.1 Model-Free and Model-Based Learning and Optimization ............................. 24
  6.2 Movement Primitives Based on Optimal Control Solutions and
      Optimization Principles ................................................................ 25
7 Inverse Optimal Control as a Method to Find Cost Functions for Humanoid
      Robot Optimization ......................................................... 27
8 Discussion and Outlook ................................................................ 32
References ................................................................................. 33

Abstract
In this chapter, we discuss optimization as a way to generate whole-body motions for humanoid robots. Optimization helps to solve many difficulties

K. Mombaur (✉)
Optimization, Robotics & Biomechanics (ORB), Institute of Computer Engineering (ZITI), Heidelberg University, Heidelberg, Germany
e-mail: katja.mombaur@ziti.uni-heidelberg.de; kmombaur@uni-hd.de

© Springer Science+Business Media B.V. 2017
A. Goswami, P. Vadakkepat (eds.), Humanoid Robotics: A Reference,
https://doi.org/10.1007/978-94-007-7194-9_33-1
related to humanoid motion generation: redundancy, feasibility, exploitation of physical capabilities, maintaining stability, as well as handling of underactuation and changing contacts with the environment. We present the formulation and numerical solution of optimal control problems for whole-body humanoid optimization with multiple phases discussing in detail different choices for objective functions and constraints to be considered. We summarize example applications of optimization for humanoid motion synthesis and motion imitation. Potential promising combinations of optimization with learning methods and movement primitives are discussed. In addition, we describe the inverse optimal control problem that helps to determine the optimality criterion underlying a recorded human motion, which then can serve as input for humanoid motion optimization. The formulation and numerical solution of these problems for locomotion examples are discussed, and example results of inverse optimal control for human locomotion based on whole-body models are shown. Further research directions in humanoid motion optimization are discussed. To give the full picture, we also mention some results for optimization in locomotion path generation and for template models which are not the focus of this chapter.

Keywords
Optimization · Optimal control · Motion generation · Motion identification · Inverse optimal control · Whole-body models · Template models · Humanoid robots

1 Introduction

Humanoid robots are complex dynamical systems for which the generation of motions is a very challenging task, since the number of parameters to tune for a motion is very high. But the challenges waiting for today’s and future humanoid robots require them to automatically generate and control a wide range of motions in order to be more flexible and adaptive to changing environments. This chapter discusses motion or trajectory optimization (which in mathematics is called optimal control) as an interesting tool to generate motions automatically based on elementary principles such as cost functions and constraints.

On the one hand, the idea to use optimization for motion generation is inspired by nature. As discussed by several authors, e.g., [90], many behaviors of humans and animals are optimal or nearly optimal. This also includes many types of motions which are assumed to be optimal due to epigenesis and phylogenesis, with the optimization criterion being adapted to the particular situation, see, e.g., [5, 6] for an optimization of mechanical aspects during locomotion. On the other hand, the use of optimization can be performance driven, if one is interested to find the best possible motion of a robot with respect to a technical criterion, such as minimum time, energy consumption, load, or maximum speed, range, or efficiency.

Optimization as a method for whole-body motion generation differs from several other methods presented in this part in the sense that it considers all system properties and goals simultaneously; motions for all individual joints are generated
at the same time. In contrast, several other methods are based on a hierarchy of evaluating goals and system properties separately (see, e.g., the stack of tasks). Optimization implicitly addresses many of the difficulties associated with humanoid motion generation:

- **Redundancy**: Humanoids – as well as humans – have many degrees of freedom (DOF) and are highly redundant systems, which means that for a given motion task, e.g., moving an object from A to B or walking on stairs of a given height, it can be executed in an infinite number of ways, involving different combinations of motions of the individual joints. Having different alternatives for a motion may be seen as an advantage, but it also presents a big challenge to motion generation since one way has to be selected over the others. Optimization can automatically do this selection since the different ways of performing the motion are judged differently from the perspective of the chosen optimization criterion.

- **Feasibility**: Despite the redundancy issue of humanoid motions, it is equally hard to find feasible motions at all. If joint motions for humanoid robots were arbitrarily picked, they would most likely result in motions which are overall kinematically or dynamically infeasible, i.e., they would violate constraints of the structure, the actuators, or the control system of the humanoid robot, result in falling, etc. Optimization solves this feasibility issue since all constraints can be directly considered in the optimal control problem formulation and will therefore be respected by the solution.

- **Underactuation**: Humanoid robots are underactuated in the general case, especially if whole-body movements such as walking are considered. Underactuation means that not every mechanical DOF of the system has a corresponding actuator. In humanoid robots, at maximum the internal DOF are actuated, but – in contrast to many mobile robots – not the overall position and orientation of the humanoid body in space. This overall motion results indirectly from the combined action of the internal joints and the contacts with the environment, e.g., the floor. The optimization approach for motion generation has no trouble addressing multibody dynamics models with less input variables than DOF and optimizes the input variables for the actuated DOF such that the overall motion (including actuated and unactuated DOF) is optimal.

- **Changing contacts**: One challenging characteristic of humanoid motions is the fact that the type, number, and intensity of contacts with the environment are permanently changing and with them the equations of motion change fundamentally. Examples are the changing foot contacts while walking, additional contacts with the hands on handrails, walls, contacts with objects during manipulation, etc. The advantages and disadvantages of changing contacts have to be taken into account by methods for motion generation. In optimization, this can be handled by formulating optimization problems with multiple phases of motion.

- **Exploitation of physical capabilities**: Optimization is also very useful to bring humanoid robots to their physical limits. Precise and realistic formulations of individual kinematic and dynamic constraints can be included in the problem, and the optimization process helps to find the solution that best exploits all
Fig. 1 Humanoid locomotion optimization can be performed on different levels: (a) whole-body walking optimization on joint level, (b) locomotion path optimization (Pictures by Martin Felis)

possibilities. Due to the all at once approach of optimization, no rules of thumb or conservative heuristics are required to maintain feasibility.

- **Stability control:** Maintaining stability throughout a motion is a particular difficulty for humanoid motion generation that does not exist for many other types of robots, such as mobile robots, crawling robots, or robot manipulators fixed to the ground. How to best define stability of anthropomorphic motion and especially how to characterize human stability are still an unsolved issue. Most humanoid robots apply specific stability criteria, e.g., based on zero moment points (ZMP), or capture points in their control strategies. These stability criteria as well as any other stability criterion, including the ones still to be defined, can be included in the constraints or objectives of the optimal control problem.

- **Human-like movements:** In humanoid robots, the goal is sometimes desired to mimic human-like movement, even though there still is no common agreement about how to precisely define this property. Optimization can be used to produce human-like motions according to some definitions, either by computing motions that follow human motion capture recordings as closely as possible or by optimizing criteria that are inspired by human behavior. The latter can be identified using the so-called inverse optimal control approach.

Solving such problems in order to achieve better whole-body motions with a particular focus on walking motions is an important task in many recent research projects, such as the European projects KoroiBot (www.koroibot.eu) and Walk-Man (www.walk-man.eu) or competitions such as the DARPA robotics challenge, and the optimization approach has received a lot of attention. Model-based optimization for robotics is also the topic of a IEEE RAS technical committee founded some years ago [84], which also has a special focus on motion and behavior optimization for humanoid robots.

As already discussed above, this chapter deals with optimization of whole-body motions for humanoid robots, i.e., the question how all the joints of the robot should be moved in a coordinated way. This corresponds to Fig. 1a. The general form of such models will be presented in Sect. 3. However, optimization obviously also
can be performed for other types of models including simpler models of humanoid motion for tasks on a higher level, as we illustrate for the example of humanoid locomotion. As shown in Fig. 1b, also the problem of humanoid walking path generation can be considered as an optimal control problem, just looking at the overall position and orientation of the robot in the plane. However, this task is not the topic of this chapter but is addressed in the chapter “Principles Underlying Locomotor Trajectory Formation” from the optimization perspective as well as from other perspectives. Once such a walking path is determined for a humanoid robot, whole-body motion generation approaches have to be used on top of it, e.g., of the type discussed in the chapter “Whole body motion planning”. A frequent way of using optimization in the context of humanoid motion generation is to apply it to template models, i.e., simple models consisting only of one or very few masses and potentially springs. Since the interest of these models is not only motion generation but also online control and they are discussed elsewhere in this book, we only briefly review related results in Sect. 2. And again, these solutions have to be post-processed by whole-body motion generation routines in order to determine all joint trajectories. We would also like to point out that this template-based approach does not address all the difficulties listed above.

In this chapter, we discuss whole-body optimal control problems for humanoid motion generation of two different types:

- Optimization for free motion synthesis using different objective functions;
- Optimization for human motion reconstruction or motion imitation of humans by humanoid robots which is related to the work discussed in the chapter “Human Motion Imitation”.

In addition, inverse optimal control is discussed as a method to identify optimality criteria of human movement, based on motion capture measurements. This chapter is not about online optimization/real-time optimal control/model predictive control which is treated elsewhere in this book (see chapter “Model Predictive Control”).

The goal of this chapter is to give fundamentals of using optimal control and underlying models to generate optimal whole-body motions for humanoid robots, summarize previous results, and provide links between optimal control and other motion generation methods. In Sect. 2, we give a brief review of the related research in the optimization of humanoid motions based on template models. In Sect. 3, we discuss the general form of multibody system models used to describe general whole-body motions for humanoid robots. Section 4 presents the formulation of multiphase motion optimization problems for humanoid robots with a particular focus on different objective functions and constraints and discusses different solution approaches. In Sect. 5, we give examples of whole-body motion optimization for humanoid robots splitting into the tasks of motion generation and motion imitation. We then discuss potential combinations of optimal control with learning and movement primitives in Sect. 6. Different methods and applications
of inverse optimal control are presented in Sect. 7. In Sect. 8, we finally give some conclusions and perspectives.

## 2 Related Research: Humanoid Optimization Based on Template Models

In this section we briefly address related research on motion optimization for template models which have also received a lot of attention, but are not the focus of this chapter. Template models describe an abstract version of a human or a robot just using one or very few bodies and define motions on the basis of very few quantities, e.g., center of mass (COM) motion, foot placement, phase timing, etc. Due to their simplicity, template models are computationally very efficient and can easily be used in optimization, also in real time. Template models often are used to study locomotion trajectories and foot placement as well some internal body characteristics such as swing foot trajectories or torso orientation (depending on the particular template model). Due to their abstract architecture, template models are suitable to describe motions for very different embodiments and are therefore good candidates for the transfer of motions between humans and humanoids. However, motions computed on the basis of template models are by no means guaranteed to satisfy all constraints of a humanoid robot. In any case, the template model results would need postprocessing to generate whole-body motions for the real robot, e.g., by whole-body planning methods as described in the chapter “Whole-body Motion Planning”.

Probably the most famous template models for motion generation and control of humanoid robots is the linear inverted pendulum or table cart model essentially using the COM of the robot, the height of which is assumed to be constant. Preview control [45] based on the table cart model is the most successful control concept which is discussed in several other places in this book. Several optimization-based approaches just using the COM are still being used with success to control humanoid robots, e.g., in the DARPA challenge for generating optimal motions on the very powerful Atlas robot [33] also planning the trajectories of the swing leg. In [34] a receding horizon controller that optimizes the next foot placement is combined with a center of mass trajectory planner that follows these footsteps and an inverse dynamics based full body controller taking into account dynamic constraints in order to produce robust dynamic walking as well as push recovery on the Atlas robot.

An extension of the COM template models is provided by models based on centroidal dynamics which additionally consider the evolution of the overall angular momentum of the system. In [23], joint trajectories are optimized using a purely kinematic model to match the COM and angular momentum of the centroidal dynamics. This requires that joint trajectories are feasible, i.e., that the actuators can produce the required torques. The authors successfully apply this method to the Atlas robot, where this assumption is satisfied for many desired motions; however for other robots it may be easily violated. Kuindersma et al. [63] perform online
trajectory planning considering the centroidal dynamics and full kinematics and track the planned trajectory using a time-varying linearization and an associated linear quadratic regulator. In [61], we use centroidal dynamics to generate multi-contact motions for the humanoid robot HRP-2 using a handrail. First, a centroidal model of the humanoid is used to find optimal contact forces as well as a kinematically feasible COM trajectory, and then whole-body motions to follow this motion are determined. By using the handrail, the power consumption of the robot could be decreased by 25%.

Other important template models include on the one hand the spring-loaded inverted pendulum (SLIP) which consists of a point mass and a spring [12, 16] as well as two-legged extensions [37], which have frequently been used to describe human running and walking. On the other hand, there are compass walkers consisting of two stiff legs [38] – essentially a double pendulum – as well as kneed extensions or quadruple pendulums. These models have been investigated a lot in the context of passive dynamic walking [74, 75] where they actually represent the full robot dynamics (see also Sect. 5), but they also serve as template models for far more complex walking systems. Many of these models have also been studied in optimization problems, e.g., to optimize stability [80, 81] or robustness [22].

As noted above, in the case of many humanoid robots, the template model-based optimization approach either risks to violate many constraints or otherwise requires a very conservative planning approach or a careful trial and error process in order to result in feasible solutions. As also discussed, in the case of very powerful robots, such as the Atlas robot, very impressive results can be more easily achieved using template models only since also challenging and fast walking motions that are generated by template models are typically within the feasible range of the dynamics.

### 3 Modeling Humanoid Robots as Multibody Systems

In this section, we discuss how humanoid robots can be modeled as multibody system models taking their essential dynamic characteristics into account. The underlying theory and algorithmic approaches to set up dynamic equations of motion are discussed elsewhere in this book. Here we summarize the general form of the equations of motion for humanoids arising for different types of movement.

In general, these models describe the DOF of the robot, geometry, mass distribution, inputs to the dynamics from actuators or even the actuator dynamics themselves, as well as passive elements in the robot. They also may include models of the control system of the robot. Even though quite complex, these models – as all models – are obviously simplifications of the real world. Figure 2 shows the abstraction from reality to model for the example of the humanoid robot HRP-2 [47] with 36 DOF. This number relates to the DOF when the robot is in the air without any contacts. This is a quite typical number of DOF for a full-sized humanoid robot; higher numbers are usually caused by more complex hand shapes, such as in the iCub (IIT) or Armar IV (KIT) robots.
As mentioned in the introduction, changing contacts, especially due to different foot and hand contacts with the environment, play an important role in humanoid motions. The changing dynamic properties can be formulated as different phases of motions. Walking, e.g., consists of a sequence of double support phases and single support phases with alternating legs for which different equations of motion have to be formulated. Also contacts with hand rails, walls, objects, other robots, etc., change the equations of motion. There are two fundamentally different ways to model contacts:

- Rigid contacts that assume that the surfaces do not deform and that there is no relative motion between the points in contact such that the number of DOF is reduced due to the contact;
- Compliant contacts that assume that contact induces forces between the bodies in contact but that there may be a deformation such that the number of DOF is not changed.

Since our experience has shown that for many humanoid applications (with existing humanoid robots) the rigid contact model results in more realistic motions, we will focus on this type here. However for walking with compliant soles or manipulations with compliant hands or palms, the other approach should be used, potentially even involving finite element models of the contact surfaces. With the adventure of new-generation robots with many soft components, also the formulation of compliant contacts in optimal control will be more frequently used.
For each motion under investigation, the different types of phases should be identified. In addition it should be analyzed if the order of phases is a priori fixed or should be kept flexible. If the contacts with the environment are modeled as inelastic, there may be discontinuities between phases at the instant when a contact occurs, if the velocities of the two points now in contact were not identical before. This possibility of velocity discontinuities should be foreseen in the model in order to have a good representation of reality; however since most humanoids prefer smooth motions (i.e., the hardware or the control systems often are not able to handle discontinuities), the generated motions in many cases will try to avoid these velocity jumps.

For the description of humanoid motions, different coordinate choices are possible. In the case of minimal coordinates, the number of coordinates would at every instant be equivalent to the number of DOF. For a system with changing contacts, this would however mean that the set of coordinates has to be changed at each contact switch. A typical coordinate choice would be to use the position and orientation of the root body of the robot (often the pelvis), and additionally all internal DOF at the joints of the humanoid which only corresponds to minimal coordinates if the robot is flying, or in compliant contact with the environment, but is otherwise redundant.

Minimal coordinates result in ordinary differential equations of the following form (with positions $q$, velocities $v = \dot{q}$, and accelerations $a = \ddot{q}$ and model parameters $p$):

$$ M(q, p)\ddot{q} + N(q, v, p) = F(q, v, p, \mathcal{M}), $$ (1)

$M$ is the humanoid’s inertia matrix which depends on the configuration of the multibody system, $N$ is the vector of nonlinear effects, and $F$ the vector of all external forces (including gravity, joint torques $\mathcal{M}$, drag, etc.). $F$ may also include the action of passive elements, e.g., of spring-damper elements.

For a redundant set of coordinates, the coupling due to the contacts can be described by a constraint of the form $g(q) = 0$ and a corresponding constraint force in the differential equation. For the coordinate choice above, this applies to all phases with some foot-ground contact or some hand contact where different types of contacts result in different sets of constraints. This results in a system of differential algebraic equations (DAE) of index 3 for the equations of motion and can be transformed into a DAE of index 1 by index reduction:

$$ \dot{q} = v $$ (2)

$$ \dot{v} = a $$ (3)

$$ \begin{pmatrix} M(q, p) & G(q, p)^T \\ G(q, p) & 0 \end{pmatrix} \begin{pmatrix} a \\ \lambda \end{pmatrix} = \begin{pmatrix} -N(q, v) + F(q, v, p, \mathcal{M}) \\ -\gamma(q, v, \mathcal{M}) \end{pmatrix} $$ (4)
with acceleration \( a = \ddot{q} \) and Lagrange multipliers \( \lambda \). The matrix \( G \) is the Jacobian of the position constraints \( G = (\partial g / \partial q) \) and \( \gamma \) the corresponding Hessian matrix \( \gamma = ((\partial G / \partial q) \dot{q}) \dot{q} \). The Lagrange multipliers are equivalent to the negative of the contact forces resulting from the corresponding constraints. The fact that ground contact during walking and running is unilateral, i.e., the ground can only push against the foot, and not pull, can be taken into account by formulating an appropriate constraint on \( \lambda \). In addition, it must be guaranteed that the original position and velocity constraints of the system are still satisfied. This is achieved by respecting the two invariant equations:

\[
\begin{align*}
    g_{\text{pos}} &= g(q(t), p) = 0 \quad (5) \\
    g_{\text{vel}} &= G(q(t), p) \cdot \dot{q}(t) = 0. \quad (6)
\end{align*}
\]

Two things should be pointed out about these equations:

- For multiple contacts, this constraint-based formulation in combination with an appropriate optimization criterion (see Sect. 4) helps to solve the redundancy issue of multibody dynamics contact formulations.
- The same equations of motions can be used to solve either the forward dynamics or the inverse dynamics problem. Only the choice of input and output variables changes, which in optimization is no fundamental difference since all variables can be left free (see below).

Changes between the different motion phases described above usually do not occur at defined time points but as a function of the state of the system. A new contact with the environment appears when the distance between any point of the robot and any point of the environment gets zero. Contact is lost when the normal contact force and thus also the corresponding Lagrange multiplier in Eq. (4) become zero. This can be described by the zero of a so-called switching function

\[
s(q(\tau), v(\tau), p) = 0. \quad (7)
\]

Assuming that impacts are fully inelastic, the velocity discontinuities can be computed as

\[
\begin{pmatrix}
    M(q, p) & G(q, p)^T \\
    G(q, p) & 0
\end{pmatrix}
\begin{pmatrix}
    v_+ \\
    \Lambda
\end{pmatrix} =
\begin{pmatrix}
    M(q)v_- \\
    0
\end{pmatrix} \quad (8)
\]

where \( v_+ \) are the unknown velocities after impact and \( v_- \) are the corresponding velocities immediately before impact. Matrices \( M \) and \( G \) are the same as in Eq. (4). Models with these two different components – the continuous phases of type (1) or (2), (3), (4), (5), and (6) – and the discrete events of type (8) are called hybrid dynamic systems.
Note that for the choice of compliant impacts, there would be no discontinuities in the velocities but only in the right-hand side of the system, and the number of DOF would remain constant, since the floor contact would be modeled by an external spring-damper force.

For describing the motions of a specific humanoid robot, the equations of motions, impact equations, and switching functions have to be established by using appropriate multibody systems formalisms. Typically they do not provide symbolic equations of motion but functions that either establish the different parts of the equations of motions, e.g., mass matrix $M$, constraint Jacobian $G$, etc., or that directly solve the forward dynamics or the inverse dynamics problem for a given input.

The automatic model generator of HuMaNS [103] uses Maple to generate explicit code to compute the entries of matrices $M$ and $G$ and vector $N$ and $y$. In our group, we have developed two different modeling tools which both are in principle suited to model general multibody systems but in our case mainly serve the primary purpose to generate whole-body human and humanoid models. The Rigid Body Dynamics Library (RBDL) [30, 32] is based on an order $n$ recursive algorithm by Featherstone [29]. The other tool, Dynamod, [52] is based on explicit code generation in a similar way as HuMaNS.

In addition, correct model parameters are required to describe the geometry and inertia properties of the robot. They usually should be known from the robot’s design process. In addition, there may be dynamic parameters linked to friction and damping in the robot as well as compliance in the joints (see above) or in external contacts which also may depend on the environmental properties. Determining correct model parameters is not always easy, and sometimes numerical parameter estimation techniques or optimum experimental design techniques might be helpful in this context.

The equations above, which describe the dynamics of the multibody system, i.e., the relationship between motion and forces/torques, can be extended by additional models describing the generation of forces/torques in particular types of actuators or the different feedback loops of the walking system.

### 4 Humanoid Motion Optimization by Solution of Optimal Control Problems

In this section, we discuss the formulation and solution of optimal control problems for the motion generation of humanoid robots, also pointing out special choices for objective functions and constraints. As mentioned in the introduction, optimal control problems of two different types will be considered: motion synthesis problems with general objective functions and motion reconstruction or motion imitation problems based on data from human motion recordings.
4.1 Formulation of Optimal Control Problems for Motion Synthesis and Motion Reconstruction

Mathematically, the task of solving an optimization problem for a model of differential equations is called an optimal control problem. Please note that in this context, the term control may refer to either open-loop or closed-loop control, depending on the fact if a control system is included in the process model or not; in many cases in fact open-loop controls, e.g., feed-forward torques, are used. The task of this problem is to simultaneously optimize trajectories and forces/torques (or controller commands) which are all unknown functions in time. For this purpose, it also does not matter if the multibody system equations are used in the forward or inverse dynamics sense which in the case of pure simulation would require to fix one type of variables in advance: in optimization, everything can be left free in both cases.

Since the underlying system is a hybrid dynamic model with multiple phases, the resulting optimal control problem is a multiphase optimal control problem with discontinuities. We assume here that we know the order of phases in the case of locomotion (which is usually true for a given gait), but that the durations of all these phases are unknown for the case of motion synthesis. A detailed overview for planning and optimizing contacts for a priori unknown orders is given in the chapter “Multi-contact Motion Planning and Control”. The general optimal control problem is formulated as

\[
\min_{x(.), u(.), p, \tau} \sum_{j=1}^{n_{ph}} \left( \int_{\tau_{j-1}}^{\tau_j} \phi_j(x(t), u(t), p) \, dt + \Phi_j(\tau_j, x(\tau_j), p) \right)
\]

s.t. \( \dot{x}(t) = f_j(t, x(t), u(t), p) \) for \( t \in [\tau_{j-1}, \tau_j] \),

\( j = 1, \ldots, n_{ph}, \tau_0 = 0, \tau_{n_{ph}} = T \)

\( x(\tau_j^+) = x(\tau_j^-) + J(\tau_j^-, x(\tau_j^-), p) \) for \( j = 1, \ldots, n_{ph} \)

\( g_j(t, x(t), u(t), p) \geq 0 \) for \( t \in [\tau_{j-1}, \tau_j] \)

\( r_{eq}(x(0), \ldots, x(T), p) = 0 \)

\( r_{ineq}(x(0), \ldots, x(T), p) \geq 0. \)

The notation in these equations is as follows: \( x(t) \) is the vector of state variables (with \( x^T = (q^T, v^T) \)), and \( u(t) \) is the vector of control or input variables of the system. In several of our examples, we are using the dynamic equations in the forward sense with torques as input, but as outlined above, this does not mean that torques have to be known in advance, but controls and states are determined simultaneously by the optimal control problem. It would also be possible to formulate the optimal control problem based on inverse dynamics, but in this case another choice of control variables would be made (e.g., equivalent to accelerations...
If the model contains a closed-loop control system, the controls $u$ could be chosen as the commands to the controllers, e.g., the trajectory to track. $p$ is the vector of free model parameters which might be, e.g., free spring or damper constants, unknown friction coefficients, adjustable geometry, inertia properties of robots, etc. $\tau$ is the vector of phase-switching times, with the overall time of the motion being $T = \tau_{n_{ph}}$. Details about the objective function and constraint choices will be discussed below.

### 4.2 Objective Functions for Humanoid Motion Synthesis

For motion synthesis, i.e., the generation of free motions for humanoid robots, many different objective functions (9) can make sense, depending on the specific task. From the mathematical perspective, we can distinguish two different types of objective function terms which are indicated in the formulation above: Lagrange-type functions that take the form of an integral over the duration of the phase and Mayer-type functions that only depend on values at the end of the respective phase. Mathematically these objective function types are equivalent to each other in the sense that transformations from one type into the other are possible by variable transformation, but for most practical problems, it is more convenient to formulate one type or the other. In the following we give some examples for objective functions without the intention to be exhaustive (equations are omitted due to reasons of space). Note that quite often combined objectives using a somehow weighted combination of some of the following are used.

Examples for general objective functions for motions:

- Minimization of energy consumption
- Minimization of joint torques
- Minimization of (absolute) mechanical work
- Maximization of efficiency
- Minimization of total time of a motion, phase times, etc.
- Maximization or minimization of characteristic velocities
- Maximization of smoothness of motion (minimization of joint accelerations or jerks)
- Minimization of interaction forces with environment/humans/other robots
- Maximization of manipulability [106]
- Minimization of deviations from periodicity
- Minimization of deviations from desired positions, angles, velocities, etc.

Examples for objective functions specific to locomotion problems (or similar whole-body problems):

- Minimization/maximization of step width/step length
- Minimization/maximization of step frequency
• Minimization of cost of transport, defined as energy used (electrical energy or mechanical energy/work) divided by weight and distance traveled
• Minimization of angular momentum about COM (about different axes)
• Maximization of stability-related criteria such as ZMP criteria (based on full model or simplified model) or capture point criteria based on a pointwise evaluation of the system
• Maximization of stability based on criteria that require evaluations of the entire trajectories/limit cycles such as Lyapunov stability criteria
• Minimization of head motions (in particular rotational head motions) to stabilize gaze.

Note that for some of these objective functions, the optimal control problem formulation given above is not sufficient and requires an augmentation. For example, stability criteria based on Lyapunov’s first method require the evaluation of sensitivities of the trajectories in the objective function and thus the integration of the hybrid variational differential equation in the problem formulation; see, e.g., [78] for details.

Examples for objective functions specific to manipulation, pointing, or grasping:

• Minimization of end point variability
• Maximization of end point velocity
• Minimization/maximization of end point acceleration
• Minimization of end point jerk

In the general case, optimal control problems are solved for free end time and phase times, but in some cases, fixed times are required.

### 4.3 Objective Function for Human Motion Imitation by Humanoids

In this paragraph, we are not interested to use optimization to improve the robot motion in terms of some of the performance measures defined above but determine a robot motion that is closest to a recorded human movement. As shown in Fig. 3, this typically involves the following steps:

• Postprocessing of motion capture data to extract recorded human kinematics from marker data, IMU data, etc.
• Kinematic motion retargeting, i.e., a transfer from human kinematics to humanoid kinematics
• Mapping this kinematic reference to a feasible humanoid robot motion.

Kinematic retargeting takes into account the different geometries, different DOF, etc., of the two systems. It is a very common task in computer graphics [58, 70].
A possible approach for this retargeting is provided by the MMM tool [99]. In humanoid robotics, also the entire process from human reference kinematics to feasible humanoid motions is sometimes called motion retargeting, also involving other techniques for the last step than discussed here (see, e.g., [77, 104, 107]).

Here, we will discuss how to solve the last step, i.e., the problem of finding a dynamically feasible robot motion that is as close as possible to corresponding reference data, by optimal control. The optimal control problem formulation remains as above, but the objective function now essentially minimizes the distance between computed motion and reference. In addition to the objective function, also the free variables change. Equation (9) is replaced by

$$\min_{x(t), u(t), p} \sum_{j=1}^{n_{ph}} \sum_{m=1}^{n_{M,j}} (x'(t_{jm}) - x_M'(t_{jm}))^T W_x (x'(t_{jm}) - x_M'(t_{jm})) + u^T W_u u.$$ (15)

This first term contains the main component of this objective function, namely, the least squares fit between measured motion and the trajectory of the model. In practice, this is often not a continuous function since measurement points are discrete. Instead, it can be computed as the sum over all measurement points, the number of which depends on measurement frequency and phase times. Typically measurement data is not available for all state variables $x$. The superscript $'$ denotes the subset of state variables that are actually measured directly or indirectly (typically the position variables, not the velocities), where $x_M'$ denotes the measured values and $x'$ the corresponding computed variables at all measurement points $t_{jm}$. $W_x$ is a diagonal weighting matrix that guarantees proper scaling of the different entries. The second term in the objective function is a regularization term slightly penalizing larger control values. The diagonal weighting matrix $W_u$ makes sure that overall, this term is very small compared to the first term and therefore does not hinder the main goal and that different control variables are properly scaled. The purpose of this term is to solve the redundancy issue in the case of multiple contact forces and to remove the effect of measurement noise on the control results. It also should be noted that in the case of motion fitting, the timing of the problem is known from the measurements, so the phase change times $\tau$ are no free variables of this problem.
4.4 Constraints to be Considered in the Optimal Control Problem

In addition to objective functions, the different constraints are important components of the optimal control problem formulation. Constraints describe all characteristics of the dynamic robot model as outlined in Sect. 3, the specific task to perform as well as details of the environment. Among the constraints to be considered are the following:

- the hybrid system dynamics equations with continuous and discrete motion phases which are described in the modeling section. For simplicity, we are not using the full multibody systems models in the optimal control problem formulation but use the simplified placeholders Eqs. (10) and (11);
- the switching constraints between the different motion phases (see (7)) which are formulated as pointwise equality constraints of type (13) at the respective phase-switching points;
- upper and lower bounds for all free variables of the optimal control problem, i.e., all controls, states, parameters, and phase times (12);
- more complex inequality constraints describing relationships between several variables, e.g., to avoid auto-collision or collision with the environment or to satisfy friction cone conditions for contacts or describe unilaterality of forces (also in (12));
- constraints related to specific requirements of the controller, e.g., in terms of stability criteria imposed by the controller, e.g., to keep the ZMP in a defined region (also of type (12));
- constraints to describe the desired motion task, among them start point constraints, end point constraints, or constraints defining via points, rest points, etc. (formulated as pointwise equality constraints of type (13));
- periodicity constraints to describe motions that are cyclic in all or part of the state variables (e.g., walking), formulated as coupled equality constraints linking states at different time points (in (13));
- additional inequality conditions at start, end, or phase-switching points (pointwise inequality constraints, in (14))
- coupled inequality constraints linking different points in time, e.g., to state relationships between positions or velocities over the course of a motion or a phase, also in (14);
- various constraints describing the environment, if necessary, e.g., definition of footholds for walking on step stones or similar, specification of possible hand contact locations, or of obstacles to avoid, etc., all resulting in constraints of type (12).

In addition, many of the objective functions described in Sect. 4.2 can equally well be used as constraints, depending on the task to be satisfied. As already indicated above, e.g., stability can either be maximized (use as criterion in objective function) or stability criteria can be kept in defined bounds (use in constraints). The same may
apply to criteria related to energy, efficiency, timing, smoothness, interaction forces, as well as many others: depending on the particular question asked, it is either more appropriate to really optimize a criterion or to constrain it to particular bounds. This split may also be used as a way to distinguish primary and secondary optimization goals.

4.5 Solution of Optimal Control Problems

The difficulty about optimal control problems as opposed to standard optimization problems is that they are infinite dimensional in the sense that the state and control variables are unknown functions in time and not just unknown $n$-dimensional vectors as in optimization problems. This requires special solution methods, and we will discuss approaches to handle the infinite dimensionality in controls and in states separately.

With respect to treating the infinite dimensionality of controls, there are three different approaches: dynamic programming and so-called direct and indirect methods.

In dynamic programming, the original continuous time problem is transformed into a discrete time problem-solving each time step separately a problem to compute the optimal control for the next interval. The state variables in this interval can then immediately be computed from that. Dynamic programming uses the principle of optimality of sub-arcs and leads to the Hamilton-Jacobi-Bellman equation [10].

In the case of the direct approach, which is also labeled “first-discretize-then-optimize” approach, the free control functions are discretized (as the name suggests), i.e., the unknown continuous functions are replaced by a finite number of free parameters. In principle, many types of discretization could be chosen, but for numerical reasons, discretizations by functions with local support, e.g., piecewise constant or piecewise linear, continuous function approximations are often preferred. The result of this discretization is a boundary value problem, i.e., the states are still infinite dimensional.

Indirect methods, which are also called “first-optimize-then-discretize” methods, are based on a formulation of the necessary optimality conditions of the infinite-dimensional optimal control problem. This also results in a boundary value problem. The well-known Pontryagin’s maximum principle [88] belongs to this type of methods. While the indirect approach is more precise since it keeps the full space of control variables, the direct method is much better suited to solve practical problems.

For both the direct and the indirect method, the solution of the boundary value problem which corresponds to the treatment of the infinite dimensionality of the state variables can be solved by shooting methods or by collocation methods [7]. The collocation method uses a discretization of states on the so-called collocation points. This results in the formulation of a finite number of constraints instead of the original continuous differential equations. Note that with this approach, the original dynamics are only explicitly satisfied at the collocation points but not in
between. In shooting methods, states are parameterized, not discretized, i.e., no accuracy is lost in this process. The methods rely on a split of tasks between an optimization process and a simulation process. The optimization process only sees and manipulates the state variables at so-called shooting points, but the full differential equations of the system are simultaneously evaluated using integrators by “shooting” integrations from these parameterized points. Single shooting only uses the parameterized state at the initial time point, and multiple shooting uses multiple points on a previously selected grid, from which independent integrations are started. Continuity conditions that have to be satisfied at the solution guarantee that the integrated trajectory pieces at the different multiple shooting intervals are continuous at the multiple shooting points. Continuity of higher-order derivatives between two intervals results automatically if the same differential equation is treated in both intervals and does not have to be imposed. It should be noted that the multiple shooting approach is also capable of treating multiphase problems as formulated above by explicitly assigning the different phase models to different sets of multiple shooting intervals. Discontinuities can be handled by assigning a multiple shooting interval of duration zero to this event in which no integration is performed, but the discontinuity equation is solved. Note also that problems with free phase times can be easily handled in multiple shooting methods by interpreting the grid as a relative grid of the phases and adding additional free parameters for the phase times. The efficient integration as well as sensitivity generation along the trajectory is an important component of every shooting method and is often the computationally most challenging part of the solution, even if efficient numerical techniques are used for integration and sensitivity computation. If the inverse dynamics formulation of the problem is chosen and accelerations are used as controls, the integration process can be very much simplified if the right discretization process is chosen. If accelerations are described as piecewise polynomials of low order, the integration to obtain velocities and positions can be performed analytically. The equations of motion are used to compute the resulting join torques. The problem is that in the case of underactuated systems, not all accelerations can be varied independently.

After these two discretization/parameterization steps described above, both direct collocation and direct shooting methods result in nonlinear programming problems (NLP), which can be solved by NLP techniques, e.g., sequential quadratic programming (SQP) methods. In both cases, the resulting matrices are highly structured due to the underlying discretization and parameterization processes, and the structures can be exploited for efficiency. Efficient direct multiple shooting methods have been developed by [13, 65] and implemented in the code MUSCOD. Houska et al. [44] developed the multiple shooting method ACADO. A direct collocation method, DIRCOL, has also been proposed [102]. Both approaches are used to generate locomotion and other types of whole-body robot motions by different researchers, but no systematic comparison of the two methods exists so far.
5 Examples of Successful Applications of Optimization for Whole-Body Motion of Humanoids and Other Anthropomorphic Systems

Many researchers have studied whole-body motions such as locomotion for anthropomorphic or simpler bipedal structures, not necessarily looking at specific humanoid robot models. Early work included the optimization of walking motions for planar bipeds, minimizing energy using direct shooting methods for a 4-link structure [91] or minimizing actuation for simple 5-link structures [40]. Our own work focused on the optimization of open-loop stability of simple bipedal walking [80] using a combination of direct multiple shooting and derivative-free methods. In [15] optimal solutions for a biped prototype robot were studied using direct collocation and implemented using trajectory tracking strategies. Bessonnet et al. [11] proposed a generation method for optimal energetic gaits with entrance and exit motion cycles based on the indirect method of Pontryagin’s maximum principle.

In [32], we have used direct multiple shooting methods to solve the nonlinear least squares optimal control problem for motion fitting in walking for a whole-dynamic model that has been previously adjusted to the subject’s geometry. We have used the same optimal control techniques to generate natural human whole-body running [94] and walking motions [31], the latter evaluating different optimization criteria. This approach was extended in [78] to optimize stability for a human running motion. Other applications of optimization in biomechanics of human movement can be found in [1, 2] where muscle activation is optimized for walking motions, also addressing the choice of gaits in lower gravity.

In computer graphics, optimization approaches for whole-body motions have been used recently [21, 85, 98]. From a robotics perspective, it is important to note that the focus here is not to get to get an exact picture of reality with precise constraints but to obtain a visually appealing visualization of a movement. The work is based on a dynamic programming techniques for motion generation of anthropomorphic motion simulation and prediction and implemented in the code MuJoCo [101] using contact smoothing algorithms [100] and complementarity relaxation and least squares treatment of virtual actuation [27]. The method is also used for whole-body model predictive control [28]. It has recently also been used to optimize whole-body motions of the humanoid robot HRP-2, also simplifying some of the unilateral contact constraints [57].

In [64], the use of optimization algorithms based on recursive dynamics using Lie algebra [87] was proposed to generate robot motions, also allowing efficient gradient evaluation. Based on this formulation, general frameworks to generate humanoid robot motions have been developed and applied to the humanoid robot HRP-2 [76, 96]. The use of optimization to generate walking under specific constraints (e.g., blocking the HRP-2 knee joint, not exceeding a given force on one leg, etc.) was demonstrated by [66, 67].

We now want to present some results on humanoid robot optimization in a bit more detail. Most of the applications discussed concern the humanoid robot HRP-
2 from Japan which has a total height of 1.58 m and weigh 58 kg and has been developed by Kawada and AIST [47]. There are several robots of this type in different labs in the world, among which one at LAAS-CNRS in Toulouse. The HRP-2 robot has 6 global and 30 internal DOF which are essentially position controlled, but it has an elasticity in the ankle joints. Even though the robot has been designed a decade ago, it is still very reliable and versatile. One of the presented results uses a model of the humanoid robot HRP-4 which is a more recent member of the HRP family, also developed by Kawada and AIST, and measures 1.51 m and weighs only 39 kg. Since the robots are much less powerful than the human and follow a different approach to control stability, it is important to use precise models of the robots to exploit all capabilities of the robot. All optimization results presented use dynamic robot models that capture all kinematic and dynamic features of the mechanics of HRP-2. But as an additional problem, HRP-2 can in fact not execute all motions that its mechanics alone would permit. The robot uses a high-level control system that is actually very conservative in its judgment and imposes strict constraints, e.g., on the trajectory of the ZMP. For formulating optimal control problems, it is important to consider also all these constraints as far as they are known. As most humanoid robots, the robot walks with full flat foot contacts, so only single support and double support have to be described for the investigation of walking motions.

Suleiman et al. [96] have studied reaching motions for the humanoid robot HRP-2 in a highly constrained environment both in front of the robot – around the reaching target – and behind. The inverse dynamics model of the HRP-2 robot was generated based on the recursive Lie algebra formalisms mentioned above. Stability bounds based on the ZMP criterion are considered during the optimization. Collision with the environment as well as self-collision is also avoided by constraints. The resulting motions for HRP-2 in the simulator – also showing the infeasible regions – as well as on the real robot are presented in Fig. 4.

We have investigated periodic walking motions for HRP-2 using direct multiple shooting, based on multiphase models of the motion, as summarized in the papers [53, 54]. Different objective functions have been studied in this context, such as the minimization of joint torques squared, maximization of forward velocity, maximization of postural stability, maximization of efficiency, and minimization of angular velocities. The criteria have a quite important impact on the style of the walking motion. Periodicity and symmetry constraints are imposed on the walking motions. In addition, we have looked at the effect of different constraints such as restrictions on the ZMP to stay very close to the ankle joint. It could be shown that relaxing the ZMP constraints generates a significantly more upright gait [53]. Figure 5 shows an example walking motion for the HRP-2 robot for gaits with variable step lengths The work has been extended using a combination between optimization, movement primitives, and learning methods; see next section.

In addition, we have looked for the largest possible step that HRP-2 can take over an obstacle, as presented in [55]. This task is very suitable for optimization, since it requires to take the robot to its extremes and exploit all motion possibilities. We could show that for such problems with very challenging dynamics, it is necessary
Fig. 4 Generation of optimal reaching motions in highly constrained environments (Courtesy of Wael Suleiman, [96])
to model all details of dynamic behavior of the robot as well as of the controllers. Ignoring, e.g., the ankle elasticity in this context leads to infeasible motions since the elasticity acts as a low-pass filter and in fact slows the motion down. It was

Fig. 5 Generation of optimal walking motions for HRP-2 in simulation and the real robot (See [18, 53, 54] for details)

Fig. 6 Computation of the largest possible step over an obstacle for the humanoid robot HRP-2
therefore not possible to play the optimal solution for the full dynamic model of the robot with all torque limits but ignoring the ankle elasticity and some control systems characteristics which had resulted in step height of 47 cm [52], but it turned out to be infeasible due to the large impact of the elasticity on the real robot. It was also not possible to simply use the primary optimization criterion which was to maximize the step height. Instead, the step height has been successively increased, and several other criteria have been optimized that were found to take all preferences of the control system into account. With this approach, it was possible to generate a new motion with record step height of 20 cm which has been executed on the real robot [52, 55] (Fig. 6).

Suleiman et al. [97] have studied the imitation of human captured motions by the humanoid HRP-2, looking in particular at a challenging boxing motion. The model is the same as described previously. Also torque and joint velocity limits are taken into account in the model. Before fitting, the data is slowed down by a transformation in time to a speed that the robot is able to satisfy. For motion imitation, the objective function penalizes deviations in the joint angle between computed model motion and target motions as well as deviations in the positions of head and hands in Cartesian coordinates, all on this new time scheme. The resulting boxing motions are shown in Fig. 7.
We also have studied motion imitation, in this case using data of a human walking available in the KoroiBot motion capture database [59]. The motion has been mapped to a model of the humanoid robot HRP-4 provided by A. Kheddar, according to the scheme discussed in Fig. 3. The postprocessing of the data to human reference model as well as the retargeting to the HRP-4 kinematics has been performed using the MMM framework [99]. The second step has been achieved using direct optimal control with a least squares objective function on all position coordinates of the robot, but not emphasizing any of the extremal points or similar. The results, as presented in [14], are shown in Fig. 8.

6 Combining Optimization with Learning and Movement Primitives for Motion Optimization

Learning methods and movement primitives represent two alternative and very popular approaches to the problem of motion generation in robotics, in addition to optimization. Even though many goals are shared with optimization, the communities and therefore the methods are still largely separated. In this section, we want to briefly outline the ideas behind the other methods and discuss the potential of combining approaches from the three different fields of optimization, learning, and movement primitives by exploiting the advantages of each approach and combining the only the best parts.

6.1 Model-Free and Model-Based Learning and Optimization

There is a strong connection between robot optimization and robot learning, in particular learning control; see, e.g., [8] for an overview. In learning control, control strategies for a motion are acquired by trial and error of a given controller. Learning control is usually performed on the real robot system but can also be applied to a model. In comparison with optimization which always works on models, especially the model-free versions of learning are always highlighted. The objectives of learning control can range from tracking given trajectories (which is the target of iterative learning control) to improving more general objectives, which in the
context of learning are described by a so-called reward function. Learning mimics the way humans achieve new movement skills, also including many failed trials. Robot learning can either take place autonomously or through guidance by a human, e.g., through learning by imitation or by demonstration. Different types of learning approaches are extensively used in robotics for applications where failure typically does not harm the robot, such as the ball in a cup example [51], manipulation of objects, such as flipping pancakes [60], or folding towels [73], tying knots [93], or wiping tables [25]. We are not aware of model-free learning approaches applied to find complex whole-body motions for humanoid robots such as walking. Problems in this case are the high dimension of the search space to be explored, the high risk to damage the robot during failed trials, as well as the big effort involved in starting each trial. The model-free learning problem has been addressed for smaller walking robots such as Leo from Delft where the search space is much smaller and the robot can sustain pushes and get up autonomously [92].

In the KoroiBot project, we have investigated the idea of combining optimal control and learning approaches. Different possibilities of combination have been proposed, and one has been pursued in more detail. For this also a detailed comparison of the outcome of the different controllers for the same model is important which has been achieved in a unified software framework. The basic idea is to start with the solution of model-based optimal control problems until a good solution is reached and the search space is narrowed down and then switch to model-free learning in order to cope with the model-reality mismatch. The research revealed the requirements for such combination to be successful, e.g., the preservation of the Markov property which is important for learning, by the online optimization solver, i.e., the nonlinear model predictive control (NMPC) method. The requirement could be satisfied, and the proposed strategy was implemented in the integrated software platform developed in the previous year. In simulations, three controllers (learning only, NMPC only, and a hybrid NMPC and RL approach) were tested for crouching and squatting motions of the Leo robot, and it could be shown that the hybrid approach was the most successful [62].

6.2 Movement Primitives Based on Optimal Control Solutions and Optimization Principles

The concept of motion or movement primitives has been originally formulated in biology, based on the idea that actions and movements of humans and animals are composed by elementary building blocks [39, 95]. Primitives can be defined temporally, in terms of elementary movements that are occurring sequentially over time, or spatially, not affecting all DOF of the system equally. The existence of motion primitives is supported by many experiments [35]. The idea of movement primitives has also been extensively used in robotics, using primitives of very different types. Movement primitives and learning are tightly linked since movement primitives are usually learned from data or sometimes from direct application to robots. Movement primitives usually represent a very efficient approach that
allows to assemble new complex movements at run time, combining the available primitives. If applied to humanoid robots, a drawback of previous movement primitives is however that they usually do not satisfy the kinematic and dynamic constraints of the robot, and therefore a posteriori adjustments would be required to make them feasible on the real robot.

In contrast to this, the whole-body motions resulting from model-based optimization as discussed in the previous sections are computationally expensive but are at the same time feasible and optimal with respect to the desired criterion. In KoroiBot, we therefore have investigated the combination of optimal control and movement primitives to combine their advantages: the general idea is to use solutions of optimal control problems for whole-body models of humanoid robots taking into account all kinematic and dynamic constraints as training data for movement primitives. In the example considered, the goal was to compute versatile walking motions with variable step length for the humanoid robot HRP-2. The computed optimal walking motions are parameterized, in the example considered by their physical step length and their step type (left or right step, cyclic, starting, stopping, or transition step). From these solutions, training data can be extracted to be fed to the learning process, where movement primitives and a mapping from gait parameters to primitive combinations are learned. For HRP-2, we have considered the angle trajectories of all actuated joints, the ZMP trajectory, and the pelvis orientation for the training data, which are sufficient quantities to control the robot. In [56] we have learned morphable movement primitives based on Gaussian processes and principal component analysis. In [18] we reformulated our morphable movement primitive model completely in a Bayesian framework, which shows significant advantages for the online motion generation phase: it allows us to generate long stepping sequences by conditioning the beginning of the current step on the end of the previous one and to determine the optimal number of primitives by approximate Bayesian model comparison, using variational free energy as an indicator for model evidence. Focusing on optimal control simulations, in [56] we showed that a small number of training motions and a small number of primitives are sufficient to generate steps with desired physical step lengths that are dynamical feasible and almost optimal. In [18] we studied the effects of different kernels and different numbers of primitives within the robot simulator OpenHRP, which gives a good prediction to the behavior on the real robot. Enhancing the kernel, improving the representation of the ZMP movement primitives, and tightening the transition conditions at the beginning/end of steps, in [20] we finally showed that the approach is feasible for the real robot HRP-2 – even for very dynamic whole-body walking motions with variable step lengths. The described combination of movement primitives and optimal control therefore allows to generate dynamically feasible and nearly optimal humanoid walking motions while reducing the computational effort for such challenging motions from the order of hours/days to the order of milliseconds/seconds.

Related work on making optimal whole-body walking motions accessible in real time has been performed by [50]. A database of optimal walking trajectories for a 3D humanoid model with 15 DOFs is built by sampling the input space in terms
of initial state and footstep sequence and solving a whole-body optimal control problem for each sample. Then a simple quadratic function is used to encode the optimized trajectories. The approach has been validated for walking on rough terrain.

Another way to combine optimization and movement primitives has been explored by [48], who developed mathematical methods to derive constants of motions and conservation laws arising from optimization principles, exploiting symmetries of geometrically invariants based on Noether’s theorem. Constants of motion in the case of Euclidean invariant trajectories express invariance to time shift, translation, and rotation. This was also used to derive movement durations and relations between movement amplitudes, durations, and drive for human movement. A first transfer of these concepts to humanoid motion generation (on the level of locomotion trajectory generation) is shown in [49].

7 Inverse Optimal Control as a Method to Find Cost Functions for Humanoid Robot Optimization

In Sect. 4.2 we have discussed several possible choices for objective functions which can be freely selected for a humanoid motion generation according to the programmer’s taste. However, another – frequently discussed – way of choosing objective functions for humanoid robot motions is to be inspired by human movement. Achieving “human-likeness” is a common goal in humanoid robotics. There is no uniform agreement about the meaning of this expression, and one way of achieving “human-like” motions is to solve the motion imitation problem discussed in Sect. 4.3, using human data. Another way is to use the optimality principle, since – as described in the introduction – human movement is assumed to be optimal or close to optimal in many cases. The idea here is to transfer the objective function that humans are using to perform a given task and which was identified based on a human model to the model of a humanoid robot to perform the same type of tasks. This concept is illustrated in Fig. 9.
This brings us to the problem of inverse optimal control: given a specific human movement for which motion capture data has been recorded and a defined model used for its description, which is the underlying objective function that gives rise to this movement? Inverse optimal control problems can be formulated for single or multiple movements or trials and single or multiple subjects.

Human movement can be quite precisely measured by different techniques, such as optical or inertial motion capture systems, force plates for ground reaction forces, EMG measurements for muscle activity, etc. These measurements allow to fully or partly reconstruct the state variable trajectories and in some cases even part of the control histories of the observed movement. The inverse optimal control problem then determines the optimization criterion that has produced the observed solution. This can be a single criterion, but usually it is a combination of different weighted criteria.

Inverse optimal control problems are hard problems since they consist in solving a parameter estimation problem within an optimal control problem. This naturally results in a bilevel formulation with the parameter identification problem in the upper level and the optimal control problem in the lower level, as shown below. Inverse optimal control typically uses a set of assumptions about potential objective function candidates which are formulated as base functions \( \Psi_i(t) \). Mathematical sets of base functions, e.g., Fourier or polynomial bases, could in principle be used for this purpose but are less interesting since the goal is to determine the objective function in terms of some physically meaningful expressions. Therefore, physically inspired base functions are used, i.e., any of the functions listed in Sect. 4.2 would qualify for this purpose. With respect to complexity, it should be noted that many of the objective functions listed break down into several base functions since there should be the possibility to consider joints or parts of the body separately (e.g., the minimization of torques squared may be expressed by base function terms relating to the torques in each joint). The choice of good base functions for a particular problem is obviously as crucial as the choice of a good dynamic model: on the one hand, the fit between optimized model and data can only be as good as the available objective function choices permit; on the other hand, the base functions must not be redundant with respect to their effect on the motion, since otherwise the inverse optimal control problem could not be solved. The different base functions \( \Psi_i(t) \) contribute to the overall objective function in a weighted sum with a priori unknown weight factors \( \alpha_i \). The inverse optimal control problem can therefore be formulated as the problem to determine the vector of weight factors \( \alpha \) that results in the best possible fit to experimental data:

\[
\min_{\alpha} \sum_{j=1}^{m} \left| x^*'(t_j; \alpha) - x'_M(t_j) \right|^2 \tag{16}
\]

where \( x^*'(t; \alpha) \) results from the solution of
Humanoid Motion Optimization

\[
\min_{x,u,p,T} \sum_{i=1}^{n} \alpha_i \int_0^T \psi_i(x(t),u(t),p)dt
\] (17)

s. t. \[
\dot{x}(t) = f(t,x(t),u(t),p) \] (18)
\[
g(t,x(t),u(t),p) \geq 0 \] (19)
\[
\varepsilon_{eq}(x(0), \ldots, x(T), p) = 0 \] (20)
\[
\varepsilon_{ineq}(x(0), \ldots, x(T), p) \geq 0. \] (21)

In the upper level, the task is to minimize the distance between the measured motion and the computed one by optimizing over the vector of weight parameters \(\alpha\). In the lower level, a forward optimal control problem is solved in every iteration of the upper level, using the current value of \(\alpha\) in order to compute the solution \(x^*\) for the evaluation of the objective function of the upper level problem. We present here a simplified formulation of the inverse optimal control problem with only single phases and ordinary differential equations, but it can without problems be extended to the type of multi-phase optimal control problems for mechanical DAEs discussed in this chapter. If a base function turns out to be irrelevant for the problem studied, the corresponding weight factor goes to zero in the solution. The problem formulation as stated above still contains one redundancy, since the solution of the lower level problem would be exactly the same if all weight factors were multiplied by the same scalar (this is true analytically, numerically the scaling might introduce slight differences). This redundancy is addressed by either fixing one of the weights or the sum of all weights to a given value. Note that the precise formulation of the fitting objective function (16) depends on the particular type of measurements chosen as well as the selected transformation method between measured quantities and model coordinates.

On the methodological side, Kalman was the first to study inverse optimal control and also to coin the term [46], but he only looked at linear problems. Later inverse optimal control has been discussed for other classes of problems, e.g. combinatorial problems [43]. Nonlinear inverse optimal control problems are a current subject of research in the mathematical optimization community, especially on reformulating the problem as a so called MPEC (mathematical program with equilibrium constraints) where essentially the optimal control problem is replaced by the corresponding first order optimality conditions [71]. Work is performed on the theoretical side (e.g. [24,105]), but also some succesful algorithms have been implemented for direct methods based on multiple shooting [41,42] and collocation [4]. In contrast to that approach, we have proposed an approach that keeps the original bilevel form [82] and works for general nonlinear inverse optimal control problems, even with multiple phases and discontinuities of the type that we are facing for studying walking motions. The method uses direct multiple shooting for the solution of the lower level optimal control problem – as described in Sect. 4.5 and implemented in the code MUSCOD – and a gradient-free optimization technique for the upper level. This choice has been made since
each upper level function evaluation involves the solution of an optimal control problem and derivative information would be hard to obtain. The derivative-free optimization code BOBYQA (Bound Optimization BY Quadratic Approximation) by Michael Powell [89] which can also handle bounds on the free parameters performs very well in this context. Recently, we have reimplemented our inverse optimal control approach based on the NLOPT versions of Powell’s BOBYQA and COBYLA codes.

In the case of human movement studies, the question of optimality and inverse optimal control has received a lot of attention. Not in all cases, formal inverse optimal control problems are solved as describe above, but often forward solutions for different individual criteria are produced and compared to data. Flash and Hogan [36] found evidence that jerk is minimized in reaching motions. Park and Levine [86] have studied different optimality criteria in running motions. Berret et al. [9] have identified combined criteria in reaching movements and [72] have studied optimality in collaborative manipulation tasks. Important related work has been performed by [69] who study realistic movement generation for character animation by physics-based models: here the problem was not to find the objective function, but to identify unknown model parameters from motion capture data using a nonlinear inverse optimization technique. In the same way as discussed in Sect. 6, there is a relationship between optimization and learning also for the inverse problem. Inverse reinforcement learning aims to tackle similar problems as inverse optimal control. As in inverse optimal control, learning systems work with an optimization objective and a performance index [3], but in learning, the performance index is typically evaluated based on real experiments and not by solving a forward optimal control problem. Sometimes learning methods are combined with optimization methods to solve problems of the inverse optimal control type (e.g., [68] and [26]). So far, no uniform finding exists about the objective function used by humans in a wide range of motions; but individual criteria for different types of motions have been discovered.

In our research, we have used inverse optimal control to study optimality criteria of different types of human movement, among them whole-body locomotion examples as shown below, but also other applications such as yo-yo playing, interaction, and locomotion paths [82]; also see the chapter “Principles Underlying Locomotor Trajectory Formation”. In [19], we have applied inverse optimal control
to walking on level ground based on a whole-body model of human walking with 14 segments and 16 DOFs in the sagittal plane (see Fig. 10a). Seven different base functions have been used for this study: four of them concern the minimization of torques squared for different parts of the body (hip, lower legs, arms, torso + head), one concerns head stabilization, and two others concern step length maximization and step frequency maximization. Data of seven trials for six different subjects was processed in parallel in order to evaluate the variation of objective functions over subjects and trials. Visibly, the walking data sets used represent different walking styles, and especially the use of arms is quite individual. For each of the motions, a set of weights could be determined, but as expected they are not identical. However, they show a significant correlation. We can therefore hypothesize that the optimization criterion of a subject for a class of motions can be seen as a combination of a general objective function term and a term describing the personal movement style. The same problem has been set up for a 3D model of human walking with 14 segments and 34 DOF. Details about the model are given in [32], and optimal solutions for different objective functions are presented in [31]. The objective functions used in inverse optimal control are the same as in the 2D case, but the fit between optimization model and data is less good than in the 2D case. In future research angular momentum terms will be added which seem to be important according to our findings in [31]. Other inverse optimal control models used to study walking include template models as discussed in Sect. 2 and as shown in Fig. 10c. This model was used to study optimality of step stone and stair walking (see [17] for details).

These inverse optimal control results have been transferred to humanoid robots in simulation: optimality criteria identified for whole-body level ground walking have been applied to a whole-body model of the humanoid robot HRP-2. Criteria identified for step-stone walking based on a template model have been transferred to corresponding template models of the HRP-2 robot and the HeiCub robot. However, one of the problems in this context is that these humanoid robots are quite limited compared to human capabilities and that the resulting solutions for the robot model hit many constraints which are far from being reached in the corresponding human movement. This may completely change the nature of the motion. We therefore have applied adjustments to the objective functions that essentially slow down the motion (by increasing the weights on time related criteria) to make it less challenging.

Our older works on inverse optimal control include the application to a 3D model of human running with 12 segments and 25 DOF [83]. The data used for this problem came from moderate pace running motions on a treadmill at a speed of 10 km/h. Objective function terms included the individual torques squared for every joint as well as a minimization of head motions and the maximization of step length. Also in this problem, we were not able to produce an optimal motion that was very close to the recorded one. This can be explained by the fact that the model used in this study did not have a sufficient number of DOF in the back and neck area. These missing mechanical DOF make it hard to follow the recorded trajectories precisely which also blurs the subsequent inverse optimal control analysis. This
drawback has been addressed in our more recent whole-body human models for walking mentioned above which are currently also used for running analysis. A summary of inverse optimal control results can also be found in [79].

8 Discussion and Outlook

In this chapter, we have discussed optimization, or more precisely optimal control, as a tool to generate whole-body motions for humanoid robots. In contrast to other approaches to motion generation discussed in this part of the handbook, optimization is an all at once approach that determines all characteristics of a motion simultaneously to satisfy all important constraints and to improve a chosen performance criterion called the objective or cost function. Optimization is therefore well suited to solve many difficulties related to humanoid motion generation, e.g., solve the redundancy problem, achieve feasible motions, control underactuated systems with changing contacts, maintain dynamic stability, and exploit the physical capabilities of the robot. Optimization is helpful to bring humanoid robots to their technical limits, i.e., it makes the most out of given robot hardware, but obviously the better the hardware, the better are the optimization results.

We have discussed the formulation of whole-body models of humanoid motions in the form of multibody systems with multiple phases of motions and potential discontinuities. We also have presented the formulation of optimal control problems for humanoid motion generation including different possible choices of objective functions and constraints which allow to generate motions of various styles and types and satisfying different goals. In addition to motion synthesis according to different types of criteria, it is also of great interest to generate human-like motions for humanoid robots. This can be either achieved by mimicking human movement minimizing a least squares fit with respect to human data, or by transferring objective functions of human movement to humanoid robots. This requires an identification of such objectives based on human data for comparable tasks, i.e., the solution of inverse optimal control problems which also has been discussed in this chapter. Inverse optimal control problems can be formulated for one or several data sets, motion types, or subjects, depending on the underlying hypothesis of how large the area of validity of the investigated optimization law is. Successful examples of using optimization for motion synthesis and motion imitation for humanoid robots from the literature as well as from our own work are discussed, including walking, obstacle crossing, boxing, and whole-body grasping tasks.

Exploring novel optimization criteria and constraint formulations and their effect on humanoid motions will be an interesting research direction for the coming years. In this context, especially the formulation of new, less conservative stability criteria than the still dominating ZMP for humanoid movement will be interesting. More dynamic forms of locomotion like fast walking ad running including phases with no contact or very small contact areas and compliant and adjustable foot shapes will require the use of more human-inspired stability criteria, also going beyond the concept of capturability. Appropriate mathematical formulations of stability and
robustness will have to be developed along with the progression in human movement understanding.

Solving whole-body optimization problem for humanoid robots with precise models and constraints as described in this chapter is a computationally challenging problem and can so far not be achieved in real time. However, it is necessary that humanoid robots can adapt quickly to new and changing environments and flexibly chose the motions that are best suited and at the same time feasible. With the very promising combination of optimal control and movement primitives discussed in Sect. 6, we have outlined one possible solution to this problem. The time-consuming optimal control problems are solved offline in this approach, and the synthesis of a motion which, e.g., flexibly takes into account possible footholds in an environment and therefore required a particular sequence of step lengths can be achieved online, generating feasible and nearly optimal motions. This seems to be a very promising research direction which will be explored further in the future.

Other approaches for online optimization briefly discussed in this chapter but also elsewhere in this book include NMPC which is slowly progressing from the application to template models to more complex models, also including phase changes. For this, not only the generation of more efficient online optimization methods is required but also of better state estimation techniques and appropriate sensors that can be used in the NMPC method. In addition, we have discussed the potential combination of optimization and learning methods in the offline and online context that allow to exploit model knowledge for generating optimal solutions for the model and then close the model-reality mismatch by continuing the learning process on the real robots. Conditions to be satisfied to make this combination work have been briefly discussed in this chapter. Future research will continue in this direction.

Additionally, further research will be performed to provide guidelines which type of model precision will be required for which kind of robot and which kind of task in the optimization context. This strongly depends on the robot hardware and the robustness of the control system. Future work will also have to put more emphasis on compliant components in the structure and the actuators.

**Acknowledgements** The research leading to these results has received funding from the EU seventh Framework Program (FP7/2007–2013) under grant agreement no 611909 (Koroibot) and the German Excellence Initiative.

**References**


68. S. Levine, V. Koltun, Guided policy search, in *ICML* (2013)


70. L. Liu, K. Yin, M. van de Panne, T. Shao, W. Xu, Sampling-based contact-rich motion control. ACM Trans. Graph. 29(4), Article 128 (2010)


