Systematic Derivation of Simplified Dynamics for Humanoid Robots

Katsu Yamane

***Abstract*—Simplified models such as the inverted pendulum model are often used in humanoid robot control because the full dynamics model of humanoid robots is too complex to design a controller. These models are usually derived from simple mechanical systems that represent the essential properties of the robot dynamics. This method for deriving simplified models is a manual process that heavily relies on the controller developer's intuition. Moreover, mapping the state and input between the original and simplified models requires model-specific code. This paper describes a general method for systematically obtaining simplified models of humanoid robots. We demonstrate an application of derived models to humanoid robot balance control using linear quadratic regulators.**

I. Introduction

Simplified models such as the inverted pendulum model are often used in humanoid robot control because it is very difficult to design a controller for the full dynamics model due to its complexity. Simplified models consist of far fewer degrees of freedom (DOF) than the full model and are also often linearized to be compatible with linear control theory.

Researchers have used a number of different simplified models to better describe the dynamics of humanoid robots. Obtaining a linear simplified model typically involves: 1) determining a simple mechanical system that represents the essential properties of the full dynamics, 2) symbolically or numerically deriving and linearizing the equation of motion of the mechanical system, and 3) defining the mapping between the simplified and full models. An issue with this process is that it must be performed manually because step 1) heavily relies on the controller developer's intuition, and step 3) requires model-specific code.

Let us consider two examples of simplified models: the one-joint inverted pendulum model (Fig. 1 left) and the two-joint inverted pendulum model (right). These models highlight the issues with conventional approach for obtaining simplified models:

- It is not always clear how to determine the parameters of the simplified model. In the two-joint inverted pendulum model, for example, intuitively the second joint should be located around the hip joint but the exact location is somewhat arbitrary.
- Computing the model states requires model-specific code. For example, the one-joint inverted pendulum model uses the center of mass (COM) of the whole body, while the two-joint inverted pendulum model requires separate COMs for the upper and lower bodies.
- Converting the input torques to the simplified models to humanoid joint torques is not straightforward because the joints of the simplified models do not directly correspond to any of the physical joints.

K. Yamane is with Disney Research, Pittsburgh, 4720 Forbes Ave. Suite 110, Pittsburgh, PA 15213. kyamane@disneyresearch.com

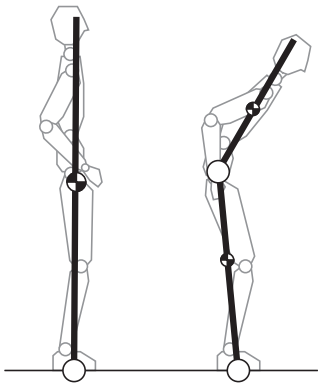

Fig. 1. One-joint (left) and two-joint (right) inverted pendulum models.

The method presented in this paper, in contrast, obtains a simplified model from only a few user-supplied information such as contact states and desired DOFs of the simplified model. The resulting model does not have an explicit mechanical structure because it is derived directly from the equation of motion of the whole humanoid body under given contact constraints. Furthermore, mapping of the state variables and inputs between the original and simplified models is derived naturally through the model simplification process and performed in the same way for any simplified models obtained by this method.

II. Related Work

Model order reduction has been studied extensively in many areas such as structural mechanics and fluid dynamics, where the original system has a very large number of generalized coordinates. For example, Lall et al. [1] presented a model reduction algorithm for flexible mechanical systems that preserves the Lagrangian structure of the original system. For graphics applications where fast and interactive simulation is critical, model reduction techniques for fluids [2] and elastostatic objects [3] have been developed.

In humanoid robot control, many simplified models have been developed based on researchers' intuition on the full-body dynamics. The simplest model that is commonly considered to represent the macroscopic dynamics of humanoid robots is the inverted pendulum model, where a lump mass is connected to the ground by a rotational joint [4]. Researchers have also proposed a number of extensions such as cart-table model [5], inverted pendulum with reaction wheel [6], [7], double inverted pendulum [8], and linear biped model [9].

Whether these models accurately represent the robot dynamics is not usually investigated in depth. An exception can be found in [10] where the author derived the physical relationships between a planar biped model and the Reaction Mass Pendulum model [7].

Our model reduction method is similar to that of Hyland et al. [11] where the reduced model is computed by minimizing the difference between the outputs simulated by the original and reduced models. They assume that the original dynamics model is stable in order to make the error finite, which is often not the case with free-standing humanoid robots. We instead minimize the difference between the kinetic energies to deal with this issue.

III. Model Reduction

A. Simplified Linear Model

The objective of this section is to approximate the dynamics of a humanoid robot with n degrees of freedom (DOF), possibly subject to contact constraints, by a $k(\ll n)$-DOF linear system

$$\hat{\boldsymbol{M}}\ddot{\boldsymbol{q}} + \hat{\boldsymbol{C}}\dot{\boldsymbol{q}} + \hat{\boldsymbol{G}}\boldsymbol{q} = \boldsymbol{u} \tag{1}$$

where $\boldsymbol{q} \in \boldsymbol{R}^k$ is the generalized coordinate of the simplified model, $\boldsymbol{u} \in \boldsymbol{R}^k$ is the input force to the simplified model, $\hat{\boldsymbol{M}} \in \boldsymbol{R}^{k\times k}$ is a symmetric, positive-definite inertia matrix, and $\hat{\boldsymbol{C}} \in \boldsymbol{R}^{k\times k}$ and $\hat{\boldsymbol{G}} \in \boldsymbol{R}^{k\times k}$ are constant matrices.

Because $\hat{\boldsymbol{M}}$ is positive definite, we can rewrite Eq.(1) as

$$\ddot{\boldsymbol{q}} = \hat{\boldsymbol{\Phi}}\boldsymbol{u} - \hat{\boldsymbol{\Phi}}\hat{\boldsymbol{G}}\boldsymbol{q} - \hat{\boldsymbol{\Phi}}\hat{\boldsymbol{C}}\dot{\boldsymbol{q}} \tag{2}$$

where $\hat{\boldsymbol{\Phi}} = \hat{\boldsymbol{M}}^{-1}$. We can then derive a state-space differential equation model

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{3}$$

by defining

$$\begin{aligned}
\boldsymbol{x} &= \begin{pmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \end{pmatrix} \\
\boldsymbol{A} &= \begin{pmatrix} \boldsymbol{0}_{k\times k} & \boldsymbol{1}_{k\times k} \\ -\hat{\boldsymbol{\Phi}}\hat{\boldsymbol{G}} & -\hat{\boldsymbol{\Phi}}\hat{\boldsymbol{C}} \end{pmatrix} \\
\boldsymbol{B} &= \begin{pmatrix} \boldsymbol{0}_{k\times k} \\ \hat{\boldsymbol{\Phi}} \end{pmatrix}
\end{aligned}$$

where $\boldsymbol{0}_*$ and $\boldsymbol{1}_*$ represent the zero and identity matrices of the size indicated by the subscript.

This model is usually obtained by symbolically or numerically evaluating and differentiating the Lagrangian equation of motion of a simple mechanical model that represents the essential properties of the full dynamics. In the following subsections, we describe a more systematic method for obtaining a simplified model (2).

B. Full Dynamics Model with Contact Constraints

We first review the original full-body dynamics of a humanoid robot subject to contact constraints.

Consider a humanoid robot with n DOF, 6 of which correspond to the translation and rotation of the base body and therefore are not actuated. Also consider the case where one or more links of the robot are in contact with the environment, enforcing m independent constraints. The dynamics of the humanoid robot is described by

$$\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{c}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) + \boldsymbol{g}(\boldsymbol{\theta}) = \boldsymbol{S}^T\boldsymbol{\tau} + \boldsymbol{J}_C^T(\boldsymbol{\theta})\boldsymbol{f}_C \tag{4}$$

where

$$\begin{aligned}
\boldsymbol{\theta} \in \boldsymbol{R}^n &: \text{generalized coordinates} \\
\boldsymbol{\tau} \in \boldsymbol{R}^{n-6} &: \text{active joint torques} \\
\boldsymbol{f}_C \in \boldsymbol{R}^m &: \text{contact forces} \\
\boldsymbol{M}(\boldsymbol{\theta}) \in \boldsymbol{R}^{n\times n} &: \text{joint-space inertia matrix} \\
\boldsymbol{c}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) \in \boldsymbol{R}^n &: \text{centrifugal and Coriolis forces} \\
\boldsymbol{g}(\boldsymbol{\theta}) \in \boldsymbol{R}^n &: \text{gravitational force} \\
\boldsymbol{J}_C(\boldsymbol{\theta}) \in \boldsymbol{R}^{m\times n} &: \text{contact Jacobian matrix}
\end{aligned}$$

and $\boldsymbol{S}^T \in \boldsymbol{R}^{n\times(n-6)}$ is the matrix that converts active joint torques to generalized forces and typically has the form

$$\boldsymbol{S}^T = \begin{pmatrix} \boldsymbol{0}_{6\times(n-6)} \\ \boldsymbol{1}_{(n-6)\times(n-6)} \end{pmatrix}. \tag{5}$$

The contact constraint is represented as

$$\boldsymbol{J}_C\ddot{\boldsymbol{\theta}} + \dot{\boldsymbol{J}}_C\dot{\boldsymbol{\theta}} = \boldsymbol{0}_m. \tag{6}$$

We can obtain the relationship between the generalized acceleration $\ddot{\boldsymbol{\theta}}$ and active joint torques $\boldsymbol{\tau}$ by the following procedure:

1) solve Eq.(4) for $\ddot{\boldsymbol{\theta}}$ and plug the result into Eq.(6)
2) solve the resulting equation for $\boldsymbol{f}_C$
3) plug $\boldsymbol{f}_C$ back into Eq.(4)

which results in

$$\ddot{\boldsymbol{\theta}} = \boldsymbol{\Phi}(\boldsymbol{\theta})\boldsymbol{S}^T\boldsymbol{\tau} + \boldsymbol{\phi}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) \tag{7}$$

where

$$\begin{aligned}
\boldsymbol{\Phi}(\boldsymbol{\theta}) &= \boldsymbol{M}^{-1} - \boldsymbol{M}^{-1}\boldsymbol{J}_C^T(\boldsymbol{J}_C\boldsymbol{M}^{-1}\boldsymbol{J}_C^T)^{-1}\boldsymbol{J}_C\boldsymbol{M}^{-1} \\
\boldsymbol{\phi}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) &= -\boldsymbol{M}^{-1}\boldsymbol{J}_C^T(\boldsymbol{J}_C\boldsymbol{M}^{-1}\boldsymbol{J}_C^T)^{-1}\dot{\boldsymbol{J}}_C\dot{\boldsymbol{\theta}} - \boldsymbol{\Phi}(\boldsymbol{c}+\boldsymbol{g}).
\end{aligned}$$

Equation (7) describes the relationship between active joint torques and joint accelerations (including the base body) assuming that the contact constraints are satisfied. Note that while $\boldsymbol{M}$ is positive definite, $\boldsymbol{\Phi}$ is only positive semi-definite due to the contact constraints, and therefore we cannot compute $\boldsymbol{\Phi}^{-1}$. This property corresponds to the fact that $\ddot{\boldsymbol{\theta}}$ that violates the contact constraints cannot be generated by any generalized force.

C. Linearized Full Dynamics Model

Because our goal is to derive a simplified linear model, we first linearize the original nonlinear model (7) around a given nominal state $(\boldsymbol{\theta}^T\ \dot{\boldsymbol{\theta}}^T)^T = (\boldsymbol{\theta}_0^T\ \boldsymbol{0}_n^T)^T$. If we define $\boldsymbol{\tau}_0$ as the joint torques required to maintain the nominal pose, $\boldsymbol{\tau}_0$ satisfies

$$\boldsymbol{0}_n = \boldsymbol{\Phi}_0\boldsymbol{S}^T\boldsymbol{\tau}_0 + \boldsymbol{\phi}_0. \tag{8}$$

where $\boldsymbol{\Phi}_0 = \boldsymbol{\Phi}(\boldsymbol{\theta}_0)$ and $\boldsymbol{\phi}_0 = \boldsymbol{\phi}(\boldsymbol{\theta}_0, \boldsymbol{0}_n)$.

When the pose and velocity change by a small amount $\delta\boldsymbol{\theta}$ and $\delta\dot{\boldsymbol{\theta}}$ respectively, the equation of motion becomes

$$\delta\ddot{\boldsymbol{\theta}} = \boldsymbol{\Phi}(\boldsymbol{\theta}_0 + \delta\boldsymbol{\theta})\boldsymbol{S}^T\boldsymbol{\tau} + \boldsymbol{\phi}(\boldsymbol{\theta}_0 + \delta\boldsymbol{\theta}, \delta\dot{\boldsymbol{\theta}}). \tag{9}$$

Defining $\boldsymbol{\tau} = \boldsymbol{\tau}_0 + \delta\boldsymbol{\tau}$, using Eq.(8), and omitting second-order terms of small changes, we obtain

$$\begin{aligned} \delta\ddot{\boldsymbol{\theta}} &= \boldsymbol{\Phi}_0\boldsymbol{S}^T\delta\boldsymbol{\tau} + (\boldsymbol{\Phi}'_\tau + \boldsymbol{\phi}'_p)\delta\boldsymbol{\theta} + \boldsymbol{\phi}'_v\delta\dot{\boldsymbol{\theta}} \\ &= \boldsymbol{\Phi}_0\boldsymbol{S}^T\delta\boldsymbol{\tau} + \boldsymbol{\Gamma}\delta\boldsymbol{\theta} + \boldsymbol{\Lambda}\delta\dot{\boldsymbol{\theta}} \end{aligned} \tag{10}$$

where $\boldsymbol{\Phi}'_\tau = \partial(\boldsymbol{\Phi}\boldsymbol{S}^T\boldsymbol{\tau}_0)/\partial\boldsymbol{\theta}$, $\boldsymbol{\phi}'_p = \partial\boldsymbol{\phi}/\partial\boldsymbol{\theta}$, and $\boldsymbol{\phi}'_v = \partial\boldsymbol{\phi}/\partial\dot{\boldsymbol{\theta}}$.

D. Model Reduction

We now describe the model reduction process consisting of two steps: 1) compute $\hat{\boldsymbol{\Phi}}$ by singular value decomposition (SVD) of $\boldsymbol{\Phi}$, and 2) obtain $\hat{\boldsymbol{G}}$ and $\hat{\boldsymbol{C}}$ by computing $\boldsymbol{\tau}$ at various poses around a nominal pose.

1) Computing $\hat{\boldsymbol{\Phi}}$: Because $\boldsymbol{\Phi}_0$ is symmetric and positive semi-definite, its SVD results in

$$\boldsymbol{\Phi}_0 = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^T \tag{11}$$

where $\boldsymbol{U} \in \boldsymbol{R}^{n\times n}$ is an orthogonal matrix containing the singular vectors and $\boldsymbol{\Sigma} \in \boldsymbol{R}^{n\times n}$ is a diagonal matrix whose diagonal elements σ_i $(i = 1, 2, \ldots, n)$ are the singular values of $\boldsymbol{\Phi}_0$ and are sorted in the descending order $(\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0)$.

Suppose we approximate $\boldsymbol{\Phi}_0$ by selecting k non-zero diagonal elements of $\boldsymbol{\Sigma}$ and corresponding columns of $\boldsymbol{U}$, such that

$$\boldsymbol{\Phi}_0 \approx \hat{\boldsymbol{\Phi}}_0 = \hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{U}}^T \tag{12}$$

with $\hat{\boldsymbol{U}} \in \boldsymbol{R}^{n\times k}$ and $\hat{\boldsymbol{\Sigma}} \in \boldsymbol{R}^{k\times k}$. Choosing appropriate singular values from $\boldsymbol{\Sigma}$ will be discussed later.

Plugging Eq.(12) into Eq.(10) yields

$$\delta\ddot{\boldsymbol{\theta}} = \hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{U}}^T\boldsymbol{S}^T\delta\boldsymbol{\tau} + \boldsymbol{\Gamma}\delta\boldsymbol{\theta} + \boldsymbol{\Lambda}\delta\dot{\boldsymbol{\theta}}. \tag{13}$$

We then left-multiply $\hat{\boldsymbol{U}}^T$ to the both sides of Eq.(13), which results in

$$\hat{\boldsymbol{U}}^T\delta\ddot{\boldsymbol{\theta}} = \hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{U}}^T\boldsymbol{S}^T\delta\boldsymbol{\tau} + \hat{\boldsymbol{U}}^T\boldsymbol{\Gamma}\delta\boldsymbol{\theta} + \hat{\boldsymbol{U}}^T\boldsymbol{\Lambda}\delta\dot{\boldsymbol{\theta}} \tag{14}$$

because $\hat{\boldsymbol{U}}^T\hat{\boldsymbol{U}} = \boldsymbol{1}_{k\times k}$.

Let us define a mapping from the generalized coordinates of the simplified model to those of the original model by

$$\delta\boldsymbol{\theta} = \hat{\boldsymbol{U}}\boldsymbol{q}. \tag{15}$$

A possible inverse mapping is

$$\boldsymbol{q} = \hat{\boldsymbol{U}}^T\delta\boldsymbol{\theta}. \tag{16}$$

We then consider the power that the inputs to the original model produce:

$$\delta\dot{\boldsymbol{\theta}}^T\boldsymbol{S}^T\boldsymbol{\tau} = \dot{\boldsymbol{q}}^T\hat{\boldsymbol{U}}^T\boldsymbol{S}^T\delta\boldsymbol{\tau} \tag{17}$$

which indicates that we can match this power to that of the simplified model by mapping the inputs by

$$\boldsymbol{u} = \hat{\boldsymbol{U}}^T\boldsymbol{S}^T\delta\boldsymbol{\tau} \tag{18}$$

Plugging them into Eq.(14), we obtain

$$\ddot{\boldsymbol{q}} = \hat{\boldsymbol{\Sigma}}\boldsymbol{u} + \hat{\boldsymbol{U}}^T\boldsymbol{\Gamma}\delta\boldsymbol{\theta} + \hat{\boldsymbol{U}}^T\boldsymbol{\Lambda}\delta\dot{\boldsymbol{\theta}}. \tag{19}$$

Note that the right-hand side of this equation has the same form as Eq.(2) with

$$\hat{\boldsymbol{M}}^{-1} = \hat{\boldsymbol{\Phi}} = \hat{\boldsymbol{\Sigma}} \tag{20}$$

$$\hat{\boldsymbol{G}} = -\hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{U}}^T\boldsymbol{\Gamma}\hat{\boldsymbol{U}} \tag{21}$$

$$\hat{\boldsymbol{C}} = -\hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{U}}^T\boldsymbol{\Lambda}\hat{\boldsymbol{U}} \tag{22}$$

because $\hat{\boldsymbol{\Sigma}}$ is a diagonal matrix with positive elements.

2) Computing $\hat{\boldsymbol{G}}$ and $\hat{\boldsymbol{C}}$: We employ a sampling-based numerical method for computing these matrices.

We first obtain $\hat{\boldsymbol{G}}$ by sampling a number of poses around $\boldsymbol{\theta}_0$ and computing the joint torques for realizing $\dot{\boldsymbol{\theta}} = \boldsymbol{0}_n$ and $\ddot{\boldsymbol{\theta}} = \boldsymbol{0}_n$ at each sample. Let us denote the difference between $\boldsymbol{\theta}_0$ and the i-th sample pose by $\Delta\boldsymbol{\theta}_i$. Using an inverse kinematics algorithm, we modify $\Delta\boldsymbol{\theta}_i$ to $\Delta\boldsymbol{\theta}'_i$ so that the pose $\boldsymbol{\theta}_0 + \Delta\boldsymbol{\theta}'_i$ satisfies the contact constraints. We then compute the joint torques $\boldsymbol{\tau}_i$ required to produce zero accelerations.

Using the mapping equations (16)(18), we convert the above quantities to those of the simplified model and plug them into Eq.(1), obtaining

$$\hat{\boldsymbol{G}}\hat{\boldsymbol{U}}^T\Delta\boldsymbol{\theta}'_i = \hat{\boldsymbol{U}}^T\boldsymbol{S}^T\boldsymbol{\tau}_i. \tag{23}$$

By collecting $\Delta\boldsymbol{\theta}'_i$ and $\boldsymbol{\tau}_i$ from a number of samples and solving a linear equation in the elements of $\hat{\boldsymbol{G}}$, we obtain $\hat{\boldsymbol{G}}$ that best fits the samples.

$\hat{\boldsymbol{C}}$ can be computed by a similar process by sampling $\Delta\dot{\boldsymbol{\theta}}$ as well as $\Delta\boldsymbol{\theta}$. Instead of zero accelerations, we compute the accelerations that satisfy the kinematic constraints (6) and compute the corresponding joint torques. Because $\hat{\boldsymbol{\Phi}}$ and $\hat{\boldsymbol{G}}$ are already known, we can form a linear equation in the elements of $\hat{\boldsymbol{C}}$.

Once we obtain $\hat{\boldsymbol{\Sigma}}$, $\hat{\boldsymbol{G}}$, and $\hat{\boldsymbol{C}}$, we can derive the linear state-space model (3).

3) Selecting the Singular Values: A standard method for dimensionality reduction is to keep the larger singular values of $\boldsymbol{\Sigma}$ and corresponding singular vectors of $\boldsymbol{U}$. In our case, however, this method does not preserve the essential properties of the original dynamics as illustrated below.

The dynamics of a physical system is often characterized by the kinetic energy. Because $\boldsymbol{\Phi}_0$ is the inverse inertia matrix of the constrained system and is singular, we first define $\bar{\boldsymbol{\Phi}}_0$ as

$$\bar{\boldsymbol{\Phi}}_0 = \bar{\boldsymbol{U}}\bar{\boldsymbol{\Sigma}}\bar{\boldsymbol{U}}^T \tag{24}$$

where $\bar{\boldsymbol{\Sigma}}$ is a diagonal matrix whose diagonal elements are the nonzero singular values of $\boldsymbol{\Phi}_0$, and $\bar{\boldsymbol{U}}$ is the matrix composed of the singular vectors corresponding to the nonzero singular values. We can then define the inverse of $\bar{\boldsymbol{\Phi}}_0$ as

$$\bar{\boldsymbol{\Phi}}_0^{-1} = \bar{\boldsymbol{U}}\bar{\boldsymbol{\Sigma}}^{-1}\bar{\boldsymbol{U}}^T \tag{25}$$

which is essentially the inertia matrix of the constrained system, because the robot cannot move in the directions of

the singular vectors of the zero singular values. The kinetic energy is therefore

$$T = \frac{1}{2}\dot{\boldsymbol{\theta}}^T \bar{\boldsymbol{\Phi}}_0^{-1} \dot{\boldsymbol{\theta}} \tag{26}$$

while the kinetic energy of the simplified model is

$$\begin{aligned} \hat{T} &= \frac{1}{2}\dot{\boldsymbol{q}}^T \hat{\boldsymbol{M}} \dot{\boldsymbol{q}} \\ &= \frac{1}{2}\dot{\boldsymbol{q}}^T \hat{\boldsymbol{\Sigma}}^{-1} \dot{\boldsymbol{q}} \\ &= \frac{1}{2}\delta\dot{\boldsymbol{\theta}}^T \hat{\boldsymbol{U}} \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{U}}^T \delta\dot{\boldsymbol{\theta}} \\ &= \frac{1}{2}\dot{\boldsymbol{\theta}}^T \hat{\boldsymbol{\Phi}}_0^{-1} \dot{\boldsymbol{\theta}} \end{aligned} \tag{27}$$

where we used Eqs.(20), (12), and (16).

In order to match $\hat{T}$ with T, we would like to make $\hat{\boldsymbol{\Phi}}_0^{-1} \approx \bar{\boldsymbol{\Phi}}_0^{-1}$ which means that we should choose the largest singular values of $\bar{\boldsymbol{\Phi}}_0^{-1}$, or in other words, the smallest nonzero singular values of $\boldsymbol{\Phi}_0$.

More formally, the singular values of $\boldsymbol{\Phi}_0$ are organized as $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_l > \sigma_{l+1} = \sigma_{l+2} = \ldots = \sigma_n = 0$ where l is the index of the last nonzero singular value. Following the above discussion, we keep the smallest k nonzero singular values (σ_{l-k+1} to σ_l) for $\hat{\boldsymbol{\Sigma}}$ and their corresponding singular vectors for $\hat{\boldsymbol{U}}$.

E. Model Reduction Results

We show an example of model reduction using a humanoid robot model with $n = 38$ DOF (32 rotational joints) and flat contacts at both feet. Figure 2 visualizes the eight most relevant singular vectors. In this case, $\boldsymbol{\Phi}$ has six zero singular values and therefore $l = 32$.

In Fig. 2, the black lines and spheres represent the link segments and joints of the humanoid robot model. The singular vector elements are represented by green and red lines starting from corresponding joints. For each rotational joint, a red line parallel to its axis is drawn with a length proportional to the magnitude of the corresponding singular vector element. If the element is negative, the line points the opposite direction of the joint axis. For the base body, red lines corresponding to the rotational degrees of freedom are drawn in the same way, while the green lines represent the elements corresponding to the linear degrees of freedom.

The left-most figure in Fig. 2 depicts the singular vector corresponding to the smallest non-zero singular value (σ_l), that is, the largest inertia. The singular vector shows that this singular vector corresponds to leaning forward. Similarly, the second to eighth smallest nonzero singular value roughly correspond to the following motions:

1) σ_{l-1}: leaning left
2) σ_{l-2}: bending the upper body to the left
3) σ_{l-3}: twisting the body to the right
4) σ_{l-4}: bending the upper body forward
5) σ_{l-5}: swinging both arms inwards
6) σ_{l-6}: swinging both arms backwards
7) σ_{l-7}: swinging both arms to the right

While these singular vectors are computed automatically, we can observe intuitive relationships with existing simplified models. For example, if we only use the first singular value (σ_l), the model would be similar to the single-inverted pendulum in the sagittal plane, while the second singular value (σ_{l-1}) would give the pendulum in the coronal plane. The third (σ_{l-2}) and fifth (σ_{l-4}) singular values correspond to the second joint of the two-joint inverted pendulum models in the sagittal and coronal planes respectively. The sixth singular value (σ_{l-5}) corresponds to changing the inertia around the COM, which is explicitly modeled in the Reaction Mass Pendulum model [7]. Finally, the seventh (σ_{l-6}) and eighth (σ_{l-7}) singular values correspond to shifting the COM without changing the contact point, which is modeled in the cart-table model [5].

On the other hand, the twist motion (σ_{l-3}) is rarely considered in simplified humanoid models but turns out to be fairly important at least in terms of the associated inertia. In the result section, we will show an example where considering this mode appears to be critical in recovering a balanced pose.

Another interesting observation is that the vertical COM motion, often considered for controlling running and jumping, does not appear among the most important modes. This is probably because the knee joints in this particular nominal pose are almost straight and therefore cannot easily generate vertical COM motion. In fact, this mode appears as the sixth mode when we compute the simplified model with bent knees. This result implies that task-specific simplified model may be obtained by simply using appropriate nominal pose.

IV. Application to Control

We demonstrate an application of simplified models derived by our method to balance control of a humanoid robot. Because one of the advantages of our method is that multiple models can be generated and used with the same code, we also demonstrate a scenario where multiple controllers are used in sequence to generate complex motions.

A. Computing the Joint Torques

Recall the input mapping

$$\boldsymbol{u} = \hat{\boldsymbol{U}}^T \boldsymbol{S}^T \delta\boldsymbol{\tau} \tag{28}$$

where the mapping from original model to simplified model is unique, but the reverse is not. We can take advantage of this property to consider other factors to determine the joint torques. This section shows an example of computing the joint torques through numerical optimization.

A controller designed for the simplified model computes a desired input $\boldsymbol{u}^*$ to the simplified model. We also usually want to consider the reference trajectories for the joints because the controller for the simplified model does not consider the individual joints. We can formulate an optimization problem for computing the joint torque $\boldsymbol{\tau}$ and the expected contact forces $\boldsymbol{f}_c$ with a cost function considering these two terms.

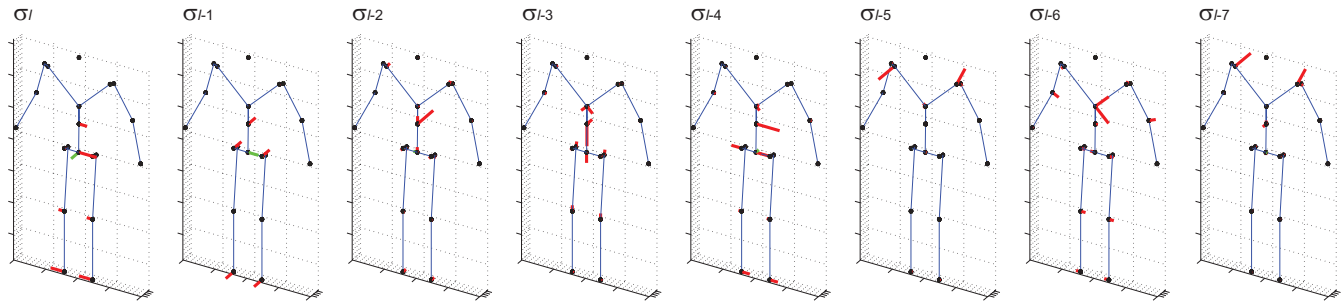

Fig. 2. Visualization of eight most relevant singular vectors.

An example of such cost function is

$$Z = \frac{1}{2}Z_u + \frac{1}{2}Z_a + \frac{1}{2}\boldsymbol{\tau}^T \boldsymbol{W}_\tau \boldsymbol{\tau}_c + \frac{1}{2}\boldsymbol{f}_c^T \boldsymbol{W}_c \boldsymbol{f}_c \qquad (29)$$

where the first and second terms address the desired input and reference trajectories respectively, and the last two terms are the damping terms with constant, positive-definite weight matrices $\boldsymbol{W}_\tau$ and $\boldsymbol{W}_c$ for the joint torque and contact forces respectively. We define Z_u as

$$Z_u = \left(\boldsymbol{u}^* - \hat{\boldsymbol{U}}^T \boldsymbol{\tau}\right)^T \boldsymbol{W}_u \left(\boldsymbol{u}^* - \hat{\boldsymbol{U}}^T \boldsymbol{\tau}\right) \qquad (30)$$

where $\boldsymbol{W}_u$ is a constant weight matrix.

For the joint trajectories, we first determine the desired joint acceleration by

$$\ddot{\boldsymbol{\theta}}^* = \ddot{\boldsymbol{\theta}}_{ref} + k_d(\dot{\boldsymbol{\theta}}_{ref} - \dot{\boldsymbol{\theta}}) + k_p(\boldsymbol{\theta}_{ref} - \boldsymbol{\theta}) \qquad (31)$$

and define Z_a as

$$Z_a = \left(\ddot{\boldsymbol{\theta}}^* - \ddot{\boldsymbol{\theta}}\right)^T \boldsymbol{W}_a \left(\ddot{\boldsymbol{\theta}}^* - \ddot{\boldsymbol{\theta}}\right) \qquad (32)$$

where $\boldsymbol{W}_a$ is a constant weight matrix.

The final optimization problem is to find $\boldsymbol{\tau}$ and $\boldsymbol{f}_c$ that minimize the cost function (29), subject to the equation of motion (4).

Optionally, we can consider constraints on joint torques to enforce the joint torque limitations. We may also want to constrain $\boldsymbol{f}_c$ such that it satisfies the constraints on vertical forces, center of pressure, and friction.

B. Controller and State Estimation

In our experiments, we use infinite-horizon linear quadratic regulators (LQRs) for controlling the simplified models, although other choices are also possible. An LQR determines the input by state feedback $\boldsymbol{u} = -\boldsymbol{K}\boldsymbol{x}$ where $\boldsymbol{K}$ is a constant gain matrix. $\boldsymbol{K}$ is determined such that the closed loop system is asymptotically stable and a cost function

$$J = \int_0^\infty \left(\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{u}^T \boldsymbol{R} \boldsymbol{u}\right) dt \qquad (33)$$

is minimized, where $\boldsymbol{Q}$ is a positive semi-definite matrix and $\boldsymbol{R}$ is a positive definite matrix.

We also apply a full-state observer to estimate the current state. The measurement is obtained by first computing the difference from the nominal pose for the controller, and then multiplying the difference by $\hat{\boldsymbol{U}}^T$. The observer gain is computed by pole assignment. Note that the state estimation does not require contact force measurement.

C. Simulation Results

We use the humanoid robot model shown in Section III-E to obtain simplified models with 2 and 5 DOFs. LQRs for each model are designed as described in Section IV-B. We use diagonal matrices for $\boldsymbol{Q}$ and $\boldsymbol{R}$. The diagonal elements of $\boldsymbol{Q}$ corresponding to $\boldsymbol{q}$ are 100 and those corresponding to $\dot{\boldsymbol{q}}$ are 1×10^{-3}. All diagonal elements of $\boldsymbol{R}$ are 1×10^{-2}. The joint feedback gains k_p and k_d in Eq.(31) are set to 16 and 8 respectively for all joints. The weight matrices for the cost function (29) are $\boldsymbol{W}_u = \boldsymbol{1}$, $\boldsymbol{W}_a = \boldsymbol{1}$, $\boldsymbol{W}_\tau = \boldsymbol{0}$, and $\boldsymbol{W}_c = 1 \times 10^{-3}$.

We also add the constraints on contact forces. The vertical forces are constrained to be positive, and the center of pressure at each foot to be in the corresponding contact area. For the friction forces, we applied a conservative constraint that the forces in x and y directions must be smaller than the normal force, which corresponds to approximating the friction cone with friction coefficient 1 by a quadrangular pyramid. We use SNOPT [12] to solve the constrained optimization problem.

Simulation results are included in the accompanying video.

1) Balancing: We first demonstrate the application to a simple balancing problem under external disturbances. We applied an external force of 200 N between $t = 0.1$ s and 0.2 s from the back to two different places: upper trunk and right shoulder. The robot is controlled by LQRs and the optimization algorithm mentioned above. We prepared two LQRs based on simplified models with two and five singular values, both linearized around an upright pose.

When an external force is applied to the trunk, the robot initially leans forward and then recovers the upright pose by moving the center of pressure (COP) towards the toe. As shown in Fig. 3, both controllers are able to bring the robot back to the equilibrium state. There is no significant difference in the performances of two controllers, although the second-order controller experiences a small collision after a brief toe contact period around $t = 0.2$. Figure 2 indicates

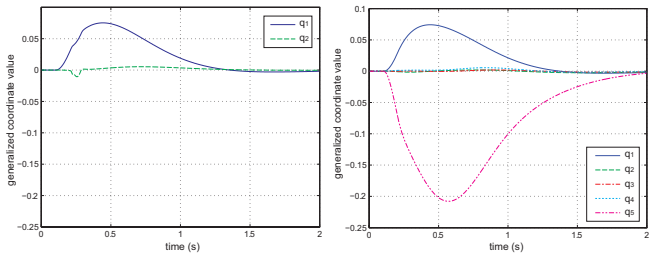

Fig. 3. Trajectories the generalized coordinates of the simplified models when pushed from the back. Left: second order, right: fifth order.

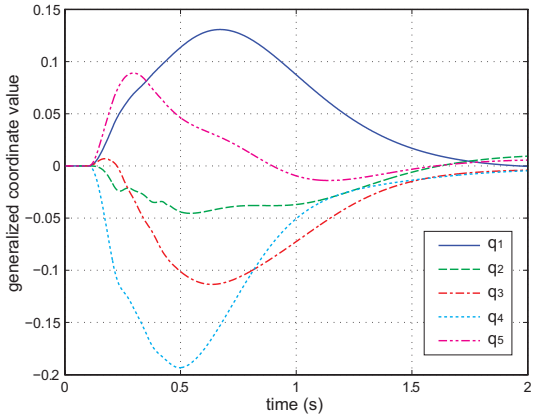

Fig. 4. Trajectories of the generalized coordinates of the simplified models when pushed at the right shoulder.

that leaning forward corresponds mostly to the first and fifth singular values, which is confirmed by the right graph of Fig. 3 where q_1 and q_5 show the largest deviation from the initial values.

An external force at the shoulder causes a twist motion, which corresponds to the fourth singular value (see Fig. 2). The controller designed based on the fifth-order model, which includes the fourth singular value, can successfully bring the robot back to the original pose as shown in Fig. 4. The snapshots from this motion are shown in Fig. 5. With the second-order model, however, the robot cannot maintain flat contacts at the feet and eventually falls down.

2) Transition to Another Pose: We can control the robot's pose by using a controller designed for a nominal pose that is different from the initial one. We generated LQRs for second- and fifth-order simplified models around a pose where the center of mass is shifted 0.11 m towards the left. The control results are shown in Figures 6 (second-order) and 7 (fifth-order), where the left graph shows the generalized coordinates from full dynamics simulation and the right graph shows the simulation of the simplified model starting from the same initial state. We can observe that the trajectories of the generalized coordinates are similar to the simplified models, which implies that the simplified model accurately represents the original dynamics.

We also generate a model at a pose with the upper torso

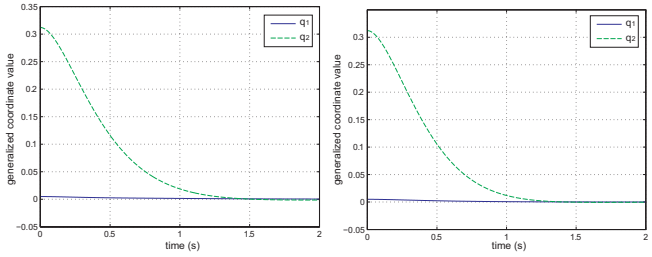

Fig. 6. Trajectories of the generalized coordinates of the second-order model during transition to a pose where the COM is shifted to the left. Left graph: full dynamics model, right graph: simplified model.

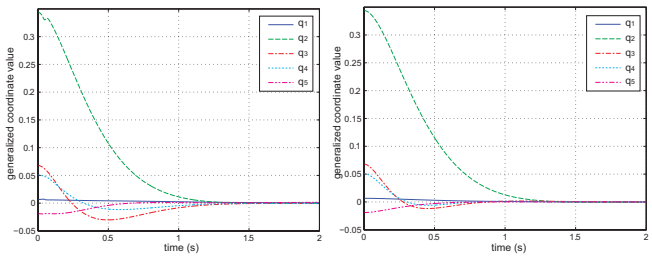

Fig. 7. Trajectories of the generalized coordinates of the fifth-order model during transition to a pose where the COM is shifted to the left. Left graph: full dynamics model, right graph: simplified model.

bent forward, which is represented by the fifth singular value. In our simulation, the second-order model cannot bring the robot to this pose, while the fifth-order model realizes the new pose as shown in Fig. 8 despite small collisions around $t = 0.2$. The snapshots from this motion are shown in Fig. 9.

3) Combining Controllers: We demonstrate that we can use multiple models and controllers with different nominal poses and contact constraints to generate more complex motions. In this particular example, we use eight fifth-order models and corrsponding controllers to generate a stepping motion. Figure 10 depicts the models used for this experiment. The top row shows the nominal pose and the first singular vector of each model, and the bottom row gives the top view of the feet and COM at each pose. Grey-colored feet have contact constraints while white-colored ones do not. The simulated robot is able to make a stepping motion

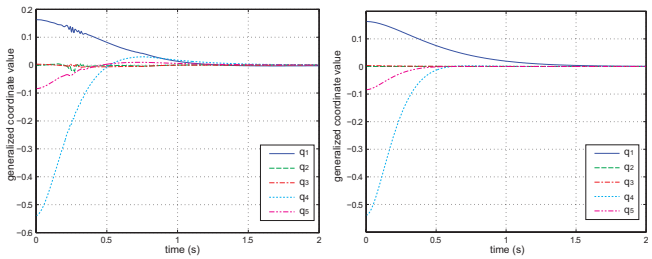

Fig. 8. Trajectories of the generalized coordinates of the fifth-order model during transition to a pose with the torso bent forward. Left graph: full dynamics model, right graph: simplified model.

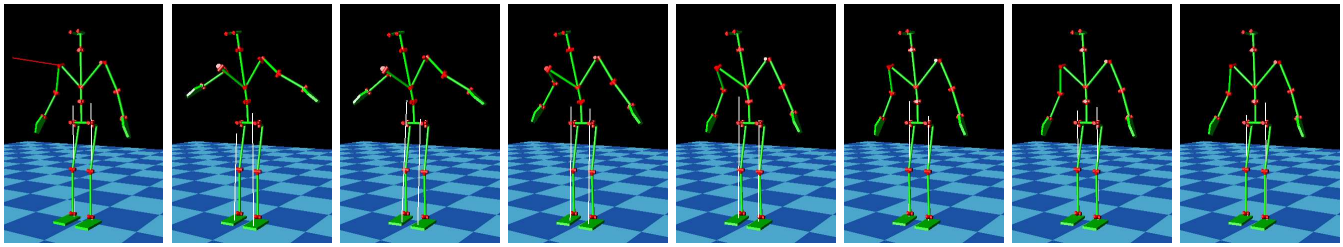

Fig. 5. Snapshots at 0.2 s interval of the motion after being pushed at the right shoulder. The white lines depict the contact forces.

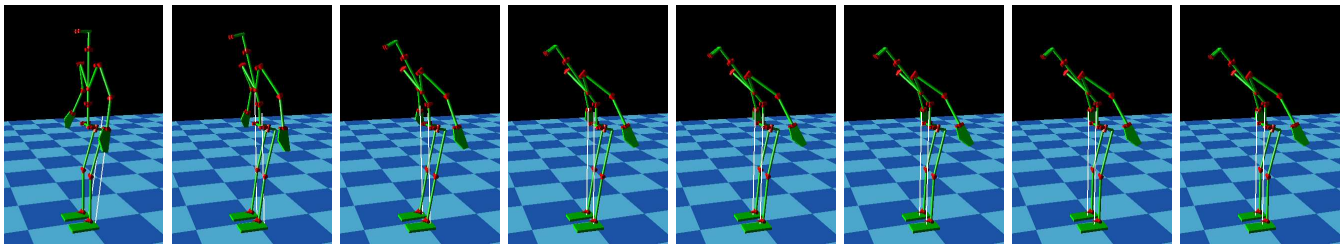

Fig. 9. Snapshots at 0.2 s interval of the transition to a pose with the upper torso bent forward.

with these controllers as shown in Fig. 11.

The conditions for switching between controllers are manually defined according to the number of contacts of current and next models. If the next model has fewer contact links than the current one (e.g., from P_1 to P_2), the switch takes place when the vertical contact force at the foot to be lifted is under 30 N for 10 consecutive frames. Similarly, if the next model has more contact links (e.g., from P_3 to P_4), the switch takes place when the vertical contact force at the foot touching down is above the same threshold (30 N) for 10 consecutive frames, in order to confirm that the new contact is established. If the same set of links are in contact at two consecutive models, the switch takes place when the total squared joint angle error falls under a certain threshold (0.05 in our experiment).

Unfortunately, the default controller does not necessarily achieve the contact force conditions for switches. To solve this problem, we take advantage of the fact that mapping from simplified model input to joint torques is not unique. If the contact force at a foot has to be smaller than the threshold to lift that foot, we start increasing the elements of $\boldsymbol{W}_c$ (see Eq.(29)) corresponding to the foot to be lifted when its vertical contact force becomes below 120 N. Similarly, if the contact force at a foot has to be larger in order to establish a contact, we start increasing the weights correspoding to the other foot when the vertical contact force at the new contact becomes over 10 N.

V. Conclusion

We presented a method for obtaining a simplified model for complex humanoid robots with contact constraints. In the conventional approach, the controller developer has to decide which mechanical model represents the essential properties of the full dynamics. Furthermore, mapping the state and input between the models often requires model-specific code. The advantages of our method include:

- It is completely automatic once the contact constraints, nominal pose, and the number of generalized coordinates of the simplified model are specified.
- We can also automatically obtain the relationship between the states and inputs of the original and simplified models. The simple linear relationship does not require any model-specific code for the mapping.
- As a result, we can easily derive the simplified models for different poses and contact constraints and switch among them using the same code. For example, we can sequentially combine multiple models and controllers to generate complex motions.

A drawback of the method is that the state and input variables of the derived simplified model do not have obvious physical meaning, and therefore it may be difficult to intuitively understand the behavior of the simplified model. Visualizing the singular vectors using figures as in Fig. 2 would give qualitative understanding of each state variable of the simplified model.

Because this method allows controller developers to generate simplified models for a number of nominal poses, it will benefit some control approaches that require multiple linear models. For example, the LQR-Tree approach [13] could be extended to more complex humanoid robots without suffering from their large dimensionality.

The method does not use external force measurement for state estimation. However, contact force data is usually available in humanoid hardware systems and may improve the accuracy if used. Furthermore, some simplified models use external forces to compute part of the state variables [14], [15]. Extension of this method to use external force information will be addressed in future work.

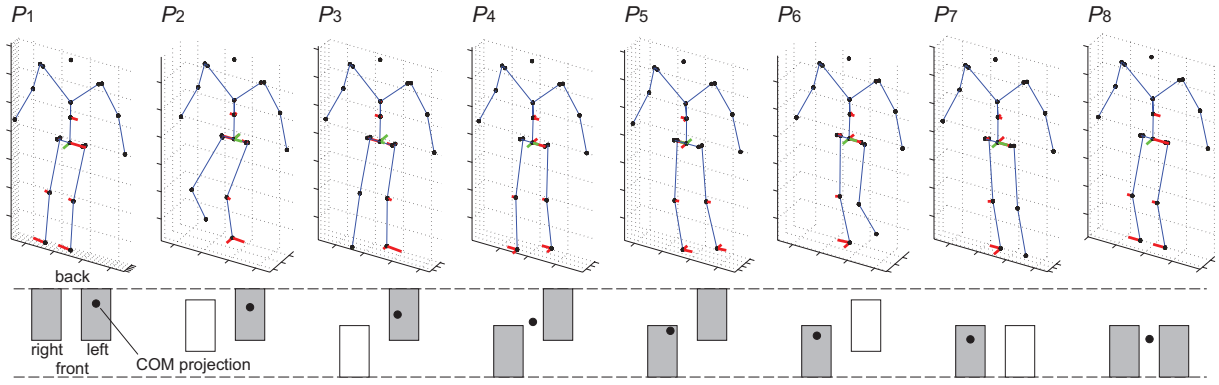

Fig. 10. The eight models used to generate the stepping example. Top row: the first singular vectors of the eight simplified models used to generate a stepping motion. Bottom row: top view of the feet and COM at the nominal pose of every model, with the foot color indicating whether contact constraints are set to the foot.

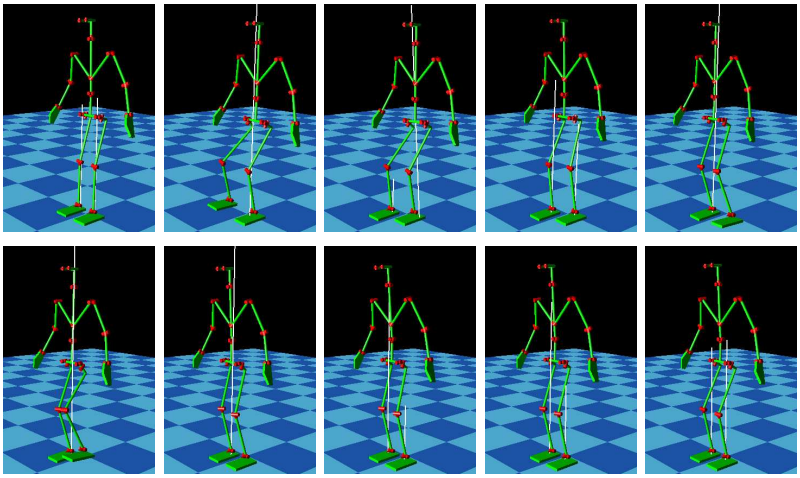

Fig. 11. Snapshots of the stepping motion at 0.8 s interval.

REFERENCES

[1] S. Lall, P. Krysl, and J. Marsden, "Structure-preserving model reduction for mechanical systems," *Physica D*, vol. 184, pp. 304–318, 2003.

[2] A. Treuille, A. Lewis, and Z. Popović, "Model reduction for real-time fluids," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 826–834, 2006.

[3] D. James and D. Pai, "Mutliresolution Green's function methods for interactive simulation of large-scale elasostatic objects," *ACM Transactions on Graphics*, vol. 22, no. 1, pp. 47–82, 2003.

[4] S. Kajita and K. Tani, "Experimental Study of Biped Dynamic Walking in the Linear Inverted Pendulum Mode," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Nagoya, Japan, May 1995, pp. 2885–2891.

[5] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2003, pp. 1620–1626.

[6] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 200–207.

[7] S. Lee and A. Goswami, "Reaction mass pendulum (RMP): An explicit model for centroidal angular momentum of humanoid robots," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2007, pp. 4667–4672.

[8] B. Stephens, "Integral control of humanoid balance," in *Proceedings of IEEE/RSJ International Conference on intelligent Robots and Systems*, 2007, pp. 4020–4027.

[9] B. Stephens and C. Atkeson, "Modeling and control of periodic humanoid balance using the liner biped model," in *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, 2009, pp. 379–384.

[10] A. Goswami, "Kinematic and dynamic analogies between planar biped robots and the Reaction Mass Pendulum (RMP) model," in *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, 2008, pp. 182–188.

[11] D. Hyland and D. Bernstein, "The optimal projection equations for model reduction and the relationships among the methods of Wilson, Skelton, and Moore," *IEEE Transactions on Automatic Control*, vol. AC-30, no. 12, pp. 1201–1211, 1985.

[12] P. Gill, W. Murray, and M. Saunders, *User's Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming*. http://www.cam.ucsd.edu/ peg/papers/sndoc7.pdf, 2006.

[13] R. Tedrake, "LQR-Trees: Feedback motion planning on sparse randomized trees," in *Proceedings of Robotics: Science and Systems*, 2009, pp. 17–24.

[14] T. Sugihara, Y. Nakamura, and H. Inoue, "Realtime Humanoid Motion Generation through ZMP Manipulation based on Inverted Pendulum Control," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington DC, May 2002, pp. 1404–1409.

[15] K. Yamane and J. Hodgins, "Simultaneous tracking and balancing of humanoid robots for imitating human motion capture data," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robot Systems*, St. Louis, MO, October 2009, pp. 2510–2517.