The MIT Humanoid Robot: Design, Motion Planning, and Control For Acrobatic Behaviors

Matthew Chignoli[1], Donghyun Kim[2], Elijah Stanger-Jones[3], and Sangbae Kim[1]

Abstract— Demonstrating acrobatic behavior of a humanoid robot such as flips and spinning jumps requires systematic approaches across hardware design, motion planning, and control. In this paper, we present a new humanoid robot design, an actuator-aware kino-dynamic motion planner, and a landing controller as part of a practical system design for highly dynamic motion control of the humanoid robot. To achieve the impulsive motions, we develop two new proprioceptive actuators and experimentally evaluate their performance using our custom-designed dynamometer. The actuator's torque, velocity, and power limits are reflected in our kino-dynamic motion planner by approximating the configuration-dependent reaction force limits and in our dynamics simulator by including actuator dynamics along with the robot's full-body dynamics. For the landing control, we effectively integrate model-predictive control and whole-body impulse control by connecting them in a dynamically consistent way to accomplish both the long-time horizon optimal control and high-bandwidth full-body dynamics-based feedback. Actuators' torque output over the entire motion are validated based on the velocity-torque model including battery voltage droop and back-EMF voltage. With the carefully designed hardware and control framework, we successfully demonstrate dynamic behaviors such as back flips, front flips, and spinning jumps in our realistic dynamics simulation.

I. INTRODUCTION

It has been three years since Boston Dynamics released the video of their humanoid robot, ATLAS [1], performing a backflip. However, no supplemental documentation was provided to answer questions regarding their methodology, the challenges they faced, or guidelines for replicating similar motions. We can speculate that the acrobatic motion is accomplished via a combination of their high powered robot and planning/control algorithms that are carefully designed based on the performance and limitations of their robot hardware. As an important step towards such demonstrations of highly dynamic motion, we present a new robot hardware design, planning, and control framework that offers practical solutions to the aforementioned questions.

The 2015 DARPA Robotics Challenge (DRC) laid the groundwork for the next generation of humanoid robots [2]–[4]. The designs of these robots and the algorithms used to control them, however, have been primarily aimed at accomplishing the minimally dynamic tasks required for the DRC [5]–[7]. The most common design paradigms for these modern humanoid robots include hydraulically actuated

[1]Department of Mechanical Engineering and [3]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA: chignoli@mit.edu [2]College of Information and Computer Sciences, University of Massachusetts Amherst, Amherst, MA 01003, USA: donghyunkim@cs.umass.edu
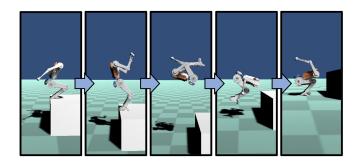
Fig. 1. **MIT Humanoid Backflip.** Proposed system designed is used to performing a back flip off of a humanoid robot off of a 0.4 m platform.

designs [8], [9] and designs that utilize physically compliant actuators [10], [11]. These actuation paradigms must contend with issues like high mechanical impedance and limited force control bandwidth, respectively, that create significant challenges for achieving acrobatic motions. The proprioceptive actuator design of the MIT Cheetah robot's has demonstrated a successful solution to this trade-off [12], but to date has not been implemented on a humanoid robot.

Manually finding acrobatic motion trajectories is nearly impossible because of the complexity of the motions and the high degree of freedom (DoF) of a humanoid robot. Consequently, trajectory optimization-based methods have emerged as the most widely used solution for the problem. Trajectory optimization-based approaches to motion planning that use full-body dynamics [13], [14] can exploit the full dynamic range of the robot, but are prone to issues like local minima and excessively long solve times. These issues can be circumvented by using a reduced-order model of the robot, like a spring-mass model [15], [16], but restrictive assumptions typically limit the generality of these approaches. Centroidal dynamics-based approaches to motion planning are popular due to their unique combination of computational tractability and dynamic expressiveness. By considering only the centroidal dynamics of the robot [17], these approaches are able to capture the core dynamics of the system without having to contend with the robot's numerous degrees of freedom. Kino-dynamic planners [18], [19] simultaneously optimize the robot's centroidal dynamics and joint-level kinematics, which offer advantages in terms of generality.

For stable landing, the robot must dissipate kinetic energy over a long time period while rapidly responding to the error in the controller's expectation caused by unmodeled dynamics, modeling error, or external disturbance. This challenge has been addressed in our previous work [20] that proposed
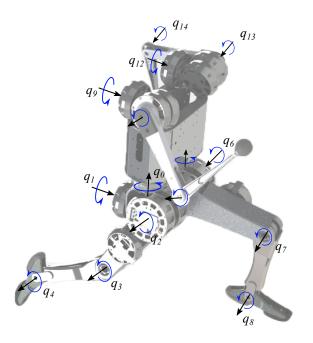
Fig. 2. **Configuration of the MIT Humanoid robot.** MIT humanoid has 5-DoF legs and 3-DoF arms. The ankle joint has only pitch directional actuation. Two contact sensors are located at toe and heel of each foot.

a hierarchical control framework integrating the model-predictive control (MPC) using a simple lumped-mass model for long-time horizon optimization and whole-body impulse control (WBIC) for instantaneous high bandwidth control. In the previous work, WBIC shared the same position command with MPC. Here, WBIC takes the optimal motion found in MPC as the position reference along with the reaction force commands to fully utilize the optimal solution. Furthermore, WBIC prioritizes a body orientation task over a centroidal momentum task to track desired centroidal angular momentum as long as the body orients command is not violated. With this carefully framed integration, we accomplish the stable landing of a humanoid robot on the ground after dynamic aerial motions.

The main contribution of this paper is a unified approach to design, motion planning, and control of a humanoid robot that enables highly dynamic motions like the back flip shown in Fig. 1. The MIT Humanoid robot is the first attempt at applying the highly successful design principles of the MIT Cheetah robots [21]–[24] to a humanoid robot. In order to leverage the full dynamic capabilities of the robot in the impulsive motions, a novel kino-dynamic planner is developed that efficiently deals with the actuator limits of the robot. For the stable landing the jumps, we utilize a hierarchical control framework by effectively integrating a model-predictive control and whole-body control.

The viability of our proposed system design is demonstrated through simulation experiments of the robot performing acrobatic motions like flips, spins, and jumps. The performance of the MIT Humanoid's custom actuators is experimentally validated and included in the simulation experiments of the robot to ensure demonstrated motions will

be feasible on the fully assembled robot.

II. System Overview

The design paradigm of previous MIT Cheetah robots has involved a unique combination of torque dense electric motors, high-bandwidth force control, and the ability to mitigate impacts through backdrivability. This design paradigm allows the robot to produce the impulses it needs to propel itself into air while also providing mechanical robustness that enables reliable control throughout the high speed impacts that occur when landing. These same design principles have been used in the design of the MIT Humanoid; a virtual model of the near-final design is shown in Fig. 2.

A. Robot Design

The MIT Humanoid will stand approximately 0.7 m tall and weigh approximately 21 kg. Approximately 75.6 % of that mass is contained in the robot's torso, shoulder, and hip, 22.5 % in the robot's legs and the remaining 1.9 % in the robot's arms. This distribution is largely invariant to the robot's pose. The legs of the robot each have five of our custom actuators: three for the hip, one for the knee, and one for the pitch of the ankle. The knee, ankle and elbow joints contain a belt gearing system to deliver higher torques at these joints. The five degrees of freedom allow each leg to produce 3D ground reaction forces at each foot as well as moments about each foot's pitch and yaw axes. The actuator setup is detailed in Fig. 2. The system will be powered from a high performance 60V, 3Ah battery pack. Four wireless contact sensors are attached on heel and toe of each foot to detect time of contact.

TABLE I
ACTUATOR SPECIFICATIONS FOR THE MIT HUMANOID.

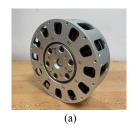| Joint	Motor Module	Gear Ratio	Max Torque	Max Joint Speed
Hip - Yaw	U10	6.0	33.6N m	55rad/s
Hip - Ab/Ad	U10	6.0	33.6N m	55rad/s
Hip - Flexion	U12	6.0	68.0N m	45rad/s
Knee	U12	12.0	136.0N m	22.5rad/s
Ankle	U10	9.33	52.2N m	35rad/s
Shoulder - Ab/Ad	U10	6.0	33.6N m	55rad/s
Shoulder - Flexion	U10	6.0	33.6N m	55rad/s
Elbow	U10	9.33	52.2N m	35rad/s

B. Custom Actuator Design

The humanoid will use a combination of two custom, high torque density electric motor modules to actuate its joints. Both modules are based on the design of the actuators originally developed for the MIT Mini Cheetah [24] but modified to meet the increased torque requirements of the MIT Humanoid. The details on the motor specifications can be found in Table I, and both modules are shown fully assembled in Fig. 3. Motor torque control is accomplished using a new version of the Mini-Cheetah motor drivers, modified to operate at the higher voltage and power levels necessary for this robot; there are no torque or force sensors.

TABLE II

MOTOR MODULE DESIGN SPECIFICATIONS

	U10 Module	U12 Module
Mass (g)	619	1174
Gear ratio	6:1	6:1
Diameter (mm)	101	120
Thickness (mm)	36	44
Maximum Torque (N m)	33.6	68.0
Maximum Speed @60V (rad/s)	55.0	45.0
Output Inertia (kg m^2)	0.0023	0.02

To validate the motors' capabilities and create an accurate model for our dynamics simulation, we constructed a custom dynamometer with a Futek TRS300 torque sensor.
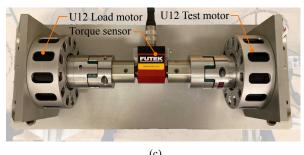

(a)   (b)

(c)

Fig. 3. **Actuator modules and dynamometer** (a) Full assembly of the U12 and (b) U10 motor modules. (c) our custom-design dynamometer.

The first step to verify the capabilities of the actuators is to measure peak torque and the torque constant using the dynamometer. Peak torque is the most critical value to ensure the model is capable of the dynamic motions in hardware. These results are detailed in Fig. 4. The U10 module reaches saturation at around 31Nm, however for the U12 saturation does not occur as the mechanical limits of the gearbox are reached first at 68Nm.

C. Actuator and Battery Dynamic Model

By combining the measured k_t value with inductance and resistance measurements taken on an impedance analyzer we can create a highly accurate model of the motors torque speed curve. The torque produced by our motor is determined by

$$\tau = \frac{3}{2} p i_q \lambda, \tag{1}$$

where τ is the produced torque, p is the number of pole pairs, i_q is the current on the q axis, and λ is the flux linkage. However the motor cannot produce arbitrary torques as the maximum current that can be put into the motor is
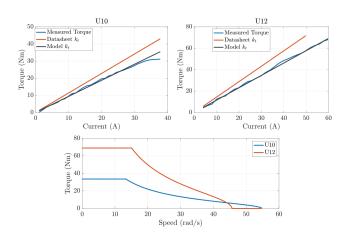

Fig. 4. (Top) Current vs Torque for for the two motor modules. (Bottom) Empirically verified torque-speed curve for the two motor modules at 60V.

dependant on the stator voltage V_s, if this exceeds the current bus voltage due to back EMF at high speeds then i_q will have to be limited, reducing torque capabilities. Note using a non-zero i_d (d-axis current - known as field-weakening) can increase torque capabilities at a reduced efficiency, however this was not necessary to achieve the trajectories in this paper. Torque speed curves for any bus voltage can be generated using

$$V_s = \sqrt{(R_s i_s)^2 + (\omega p)^2 \left((L_q i_q)^2 + (\lambda + L_d i_d)^2\right)}, \tag{2}$$

$$i_s = \sqrt{i_q^2 + i_d^2}, \tag{3}$$

where R_s is stator resistance, i_s is total stator current, ω is rotational velocity, L_q and L_d are q- and d-axis inductances. The motor actuator model also includes estimates for damping and friction coefficients.

In previous work [25], we have seen issues where the limitations of the on-board battery can cause differences between capabilities of simulation and reality; especially during highly dynamic movements. As a battery draws more power, its output voltage sags which limits the capabilities of the motors. Note voltage cannot drop further than 50 % of its fully charged state as this will greatly damage the battery. This phenomena is reflected in the relationship

$$P = \sum_{i}^{n_m} \frac{\dot{q}_i \tau_i}{\eta}, \tag{4}$$

$$V_{\text{fin}} = \frac{1}{2} \left(V_{\text{init}} + \sqrt{V_{\text{init}}^2 - 4PR_b}\right), \tag{5}$$

where R_b is the measured battery impedance, P is total power from the motors, η is the estimated motor efficiency found from stator resistance, n_m is the total number of motors, V_{init} is the bus voltage at no load and V_{fin} is the bus voltage at the given load. By combining this new voltage estimate with the torque speed curve calculations in (2) we can determine highly accurate actuator limits at every time step in simulation.
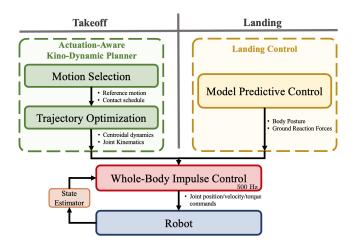
Fig. 5. **Planning and Control Framework.** Kino-dynamic planning, MPC-based landing control, and WBIC framework to achieve dynamic aerial motions of a humanoid robot.

III. PLANNING AND CONTROL

Dynamic aerial motions of the humanoid robot are achieved via the planning and control framework illustrated in Fig. 5. The motions are broken down into three phases: takeoff, flight, and landing. For takeoff, planning is carried out via the novel Actuation-Aware Kino-Dynamic Planner (AAKD). The first stage of AAKD planning is a motion selector, which, based on an evaluation of the terrain and the task at hand, selects the necessary motion for the robot (e.g. jump onto the table, do a backflip, etc.). The second stage consists of a centroidal dynamics-based trajectory optimization that finds a dynamically feasible trajectory for the robot that accomplishes the selected motion. For flight, the joints of the robot are simply controlled using PD control to a configuration determined by the AAKD planner. For landing, a quadratic programming (QP)-based MPC is used to plan reaction force profiles to stabilize the robot as it comes back into contact with the ground. In both the takeoff and landing phases, planned motions are realized via a whole-body impulse controller optimized for stabilizing motions with large rotational components.

A. Kino-Dynamic Planning

The optimization variables of the kino-dynamic optimization for a robot with n joint degrees of freedom include discrete trajectories for the generalized position vector of the robot $\mathbf{q} \in \mathbb{R}^{n+6}$, the generalized velocity vector for the robot $\dot{\mathbf{q}} \in \mathbb{R}^{n+6}$, the position $\mathbf{r} \in \mathbb{R}^3$, velocity $\dot{\mathbf{r}} \in \mathbb{R}^3$, and acceleration $\ddot{\mathbf{r}} \in \mathbb{R}^3$ of the robot's center of mass (CoM), the robot's centroidal angular momentum (CAM) $\mathbf{h} \in \mathbb{R}^3$, it's time derivative $\dot{\mathbf{h}} \in \mathbb{R}^3$, the positions of the robot's contact points $\mathbf{c}_i \in \mathbb{R}^3$, and the external ground reaction forces acting at the contact points $\mathbf{f}_i \in \mathbb{R}^3$. All n_c contacts are treated as points contacts. Each of these trajectories are discretized into n_t timesteps where n_t is the total number of timesteps and Δt is the time between each timestep. These variables are condensed into a single optimization variable

$\mathbf{X} \in \mathbb{R}^{(2n+6n_c+27) \times n_t}$ where $\mathbf{X} = [\mathbf{X}_1, ..., \mathbf{X}_{n_t}]$ and

$$\mathbf{X}_k = \begin{bmatrix} \mathbf{x}_k^\top & \mathbf{r}_k^\top & \dot{\mathbf{r}}_k^\top & \ddot{\mathbf{r}}_k^\top & \mathbf{h}_k^\top & \dot{\mathbf{h}}_k^T & \mathbf{c}_k^\top & \mathbf{f}_k^\top \end{bmatrix}^\top, \quad (6)$$

with

$$\mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_{n_c} \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_{n_c} \end{bmatrix}. \quad (7)$$

The motion selector provides the trajectory optimization with a reference motion $\mathbf{X}_{\text{ref}} \in \mathbb{R}^{(2n+6n_c+27) \times n_t}$ and a set of allowable terminal states $\mathcal{X}_{\text{term}}$. While the framework is amenable to more sophisticated approaches to motion selection, the motions demonstrated in this work all emerge from a simple motion selector that specifies a terminal state of the robot and then generates a reference motion by interpolating between the initial state and this final state. Examples of the terminal state of the robot could be that the robot ends up on top of an elevated surface, that it ends up on the other side of a wide gap, or that it rotates 360° about one of its axes.

The terminal condition $\mathbf{X}_{n_t} \in \mathcal{X}_{\text{term}}$ ensures that the output of the kino-dynamic optimization achieves the desired motion. The reference motion is used to guide the optimization towards the desired motion via the objective function

$$\min_{\mathbf{X}} \quad \sum_{k=1}^{n_t} ||\mathbf{X}_{k,\text{ref}} - \mathbf{X}_k||^2_{\boldsymbol{Q}_\mathbf{X}}, \quad (8)$$

where $\boldsymbol{Q}_\mathbf{X} \in \mathbb{R}^{(2n+6n_c+27) \times (2n+6n_c+27)}$ is a weight matrix.

The following constraints are standard for kino-dynamic optimization [18] and are enforced at every timestep k to ensure kinematic and dynamic feasibility of the generated motions. First, dynamical constraints encode the centroidal dynamics of the robot

$$m\ddot{\mathbf{r}}_k = \sum_{i=1}^{n_c} \mathbf{f}_{i,k} + m\mathbf{g}$$
$$\dot{\mathbf{h}}_k = \sum_{i=1}^{n_c} (\mathbf{c}_{i,k} - \mathbf{r}_k) \times \mathbf{f}_{i,k}, \quad (9)$$
$$\mathbf{h}_k = \boldsymbol{A}_{\text{CAM}}(\mathbf{q}_k)\dot{\mathbf{q}}_k$$

where $\boldsymbol{A}_{\text{CAM}}$ is the angular portion centroidal momentum matrix [26]. Continuity between timesteps is enforced via backward Euler integration. Kinematic feasibility is ensured via range of motion constraints on the joints as well as constraints that enforce agreement between kinematics and centroidal dynamics

$$\mathbf{r}_k = \boldsymbol{\gamma}_{\text{CoM}}(\mathbf{q}_k)$$
$$\mathbf{c}_{i,k} = \boldsymbol{\gamma}_{c_i}(\mathbf{q}_k) \quad \text{for } i = 1, 2, ..., n_c, \quad (10)$$

where $\boldsymbol{\gamma}_{\text{CoM}}$ and $\boldsymbol{\gamma}_{c_i}$ are functions that map the joint configuration of the robot to center of mass and end effector positions, respectively. The last of the standard kino-dynamic constraints are the contact constraints. Because the contact schedule is specified prior to the optimization, constraints

can be separately imposed for limbs in contact versus out of contact. Limbs in contact are subject to the constraints

$$
\begin{aligned}
\mathbf{c}_{i,k} &= \mathbf{c}_{i,k-1}, \\
c_{i,z} &= \boldsymbol{\gamma}_{\mathrm{gnd}}(c_{i,x}, c_{i,y}), \\
\mathbf{f}_{i,k} &\in \boldsymbol{\mathcal{F}},
\end{aligned}
\tag{11}
$$

where $\boldsymbol{\gamma}_{\mathrm{gnd}}$ returns the height of the ground at a given point and $\boldsymbol{\mathcal{F}}$ represents the friction cone or, in the case of this work, a linear approximation of it. Limbs out of contact are subject to

$$
c_{i,z} \geq \boldsymbol{\gamma}_{\mathrm{gnd}}(c_{i,x}, c_{i,y}).
\tag{12}
$$

A novel aspect of the proposed kino-dynamic planner is the inclusion of constraints that ensure the actuator limits of the robot are respected. Joint torques are not explicitly included in centroidal dynamics, and thus they do not show up in the kino-dynamic optimization. Mapping ground reaction forces to joint torques introduces significant nonlinearity to the formulation. For slow-moving motions like walking, actuator limits are typically dealt with via cost functions that penalize large reaction forces or constraints on maximum reaction force. In the case of highly dynamic motions that push the hardware limits of the robot, these simple techniques are not sufficient.

At a given time step, the exact joints torques required for the commanded motion are given by

$$
\boldsymbol{\tau}_j = \boldsymbol{S}_j \left(\boldsymbol{H}\ddot{\mathbf{q}} + \boldsymbol{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_G - \sum_{i=1}^{n_c} \boldsymbol{J}_{c,i}^{\top}\mathbf{f}_{c,i} \right),
\tag{13}
$$

where $\boldsymbol{S}_j \in \mathbb{R}^{n \times n+6}$ is a selection matrix for the joint dynamics of the system, $\boldsymbol{\tau}_j \in \mathbb{R}^n$ is the vector of joint torques, $\boldsymbol{H} \in \mathbb{R}^{n+6 \times n+6}$ is the mass matrix, $\boldsymbol{C}\dot{\mathbf{q}} \in \mathbb{R}^{n+6}$ is the Coriolis vector, and $\boldsymbol{\tau}_G \in \mathbb{R}^{n+6}$ is the gravity vector [27]. Empirically, we find that the reaction force terms dominate this relationship, and thus the required joints torques can be well approximated by

$$
\boldsymbol{\tau}_j \approx \boldsymbol{S}_j \left(-\sum_{i=1}^{n_c} \boldsymbol{J}_{c,i}^{T}\mathbf{f}_{c,i} \right).
\tag{14}
$$

Even with this approximation, considerable nonlinearity remains due to the dependence of the contact Jacobian on the robot's configuration. Observations of generated acrobatic motions show that the contact Jacobians of the leg are minimally sensitive to the positions of the hip yaw joints, the hip ab/ad joint, and the ankle joint. To further approximate (14), the contact Jacobians of the leg are linearized with respect to a reference position of these three joints (no linearization occurs with respect to the hip flexion or knee joints). This approximate relationship between ground reaction forces and joint torques is used in a constraint that encodes a linear approximation of the torque speed relationship in Fig 4.

B. Landing Model-Predictive Control

Once the robot detects full contact with the ground, the landing controller is activated and starts stabilizing the robot's balance by properly absorbing the impact and dissipating the kinetic energy. This requires high-bandwidth feedback control that can account for optimal motion over the long sequence of the landing motion. To accommodate both real-time computation requirement and long time period optimization, we utilize the hierarchical control framework proposed in [20] consisting of MPC and WBIC.

In this MPC formulation, a simplified lumped mass model is used to speed up the optimization process. To simplify the formulation even further, we use linear time-invariant dynamics based on the assumption that the body orientation does not change much and use Euler angle for the body orientation state representation. The Euler angle has a singularity issue, but for simplicity, we follow the convention used in [20]. Since the body orientation is assumed to be nearly constant during the MPC horizon, the angular velocity and inertia tensor are defined as follows.

$$
\dot{\boldsymbol{\Theta}}_t \approx \boldsymbol{R}_0 \boldsymbol{\omega}_t,
\tag{15}
$$

$$
_{\mathcal{G}}\boldsymbol{I} \approx \boldsymbol{R}_0 \,_{\mathcal{B}}\boldsymbol{I}\boldsymbol{R}_0^{\top},
\tag{16}
$$

where $\dot{\boldsymbol{\Theta}} = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^{\top}$ is roll (ϕ), pitch (θ), and yaw (ψ) velocity of the body, $\boldsymbol{R}_0 \in SO(3)$ is a current body orientation, which translates the angular velocity in the global frame, $\boldsymbol{\omega} \in \mathbb{R}^3$, to the local (body) coordinate, and $_{\mathcal{G}}\boldsymbol{I} \in \mathbb{R}^{3\times3}$ and $_{\mathcal{B}}\boldsymbol{I} \in \mathbb{R}^{3\times3}$ are the inertia tensor seen from the global and local (body) frame, respectively.

The final approximation used in the MPC is that one directional angular velocity is significant and off-diagonal terms of the inertia tensor are small. In most flip or turning jump cases, angular velocity about one principal axis is significantly larger than the others. For example, the pitch directional angular velocity is significant in the case of back/front flips and the yaw directional angular velocity is dominant in 180° turning jump. Based on this approximation, we ignore precession and nutation effects, $\boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\boldsymbol{I}\boldsymbol{\omega}) \approx \boldsymbol{I}\dot{\boldsymbol{\omega}}$.

With the above simplifications, the discrete dynamics of the system can be expressed as

$$
\mathbf{x}(k+1) = \boldsymbol{A}_k\mathbf{x}(k) + \boldsymbol{B}_k\hat{\mathbf{f}}(k) + \hat{\mathbf{g}},
\tag{17}
$$

where,

$$
\begin{aligned}
\mathbf{x} &= \begin{bmatrix} \boldsymbol{\Theta}^{\top} & \mathbf{p}^{\top} & \boldsymbol{\omega}^{\top} & \dot{\mathbf{p}}^{\top} \end{bmatrix}^{\top}, \\
\hat{\mathbf{f}} &= \begin{bmatrix} \mathbf{f}_1 & \cdots & \mathbf{f}_n \end{bmatrix}^{\top}, \\
\hat{\mathbf{g}} &= \begin{bmatrix} \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{g}^{\top} \end{bmatrix}^{\top},
\end{aligned}
\tag{18}
$$

$$
\boldsymbol{A} = \begin{bmatrix}
\mathbf{1}_{3\times3} & \mathbf{0}_{3\times3} & \boldsymbol{R}_0\Delta t & \mathbf{0}_{3\times3} \\
\mathbf{0}_{3\times3} & \mathbf{1}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{1}_{3\times3}\Delta t \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{1}_{3\times3} & \mathbf{0}_{3\times3} \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{1}_{3\times3}
\end{bmatrix},
$$

$$
\boldsymbol{B} = \begin{bmatrix}
\mathbf{0}_{3\times3} & \cdots & \mathbf{0}_{3\times3} \\
\mathbf{0}_{3\times3} & \cdots & \mathbf{0}_{3\times3} \\
_{\mathcal{G}}\boldsymbol{I}^{-1}[\mathbf{r}_1]_{\times}\Delta t & \cdots & _{\mathcal{G}}\boldsymbol{I}^{-1}[\mathbf{r}_n]_{\times}\Delta t \\
\mathbf{1}_{3\times3}\Delta t/m & \cdots & \mathbf{1}_{3\times3}\Delta t/m
\end{bmatrix}.
\tag{19}
$$

In the QP, we minimize the reaction forces and the error between the reference and state during the MPC horizon.

$$\min_{\mathbf{x},\mathbf{f}} \sum_{k=0}^{m} ||\mathbf{x}(k+1) - \mathbf{x}^{\text{ref}}(k+1)||_{\boldsymbol{Q}} + ||\mathbf{f}(k)||_{\boldsymbol{R}} \quad (20)$$

subject to dynamics, initial condition constraints, and the ground reaction force constraints,

$$| f_x | \le \mu f_z, \quad | f_y | \le \mu f_z, \quad f_z > 0, \quad (21)$$

where μ is the friction coefficient. The desired CoM position is defined by the center of the four contact locations of the feet (the right toe/heel and the left toe/heel) and the desired height is 0.05 m higher than the CoM height at the landing. The desired body orientation is also determined based on the landing posture but the pitch angle is adjusted to be smaller than 0.8 rad. The time step used in MPC is 0.1 s and the length of horizon is 15. Therefore, MPC optimizes 1.5 s future state in one iteration. The update frequency of MPC is 10 Hz, which is slower than our real-time control frequency of 500 Hz, while WBIC runs at the real-time frequency to obtain high-bandwidth posture control.

C. Task Setup of Whole-Body Impulse Control

The full formulation of WBIC is explained in [20], and its overall framework is similar to other WBCs but has a feature to include the reference reaction forces coming from the planner or the MPC, which we explained in the previous sections. The performance of WBIC has been validated by showing the high-speed running of a quadruped robot, but two new important issues arise in the landing control. Note that these issues are specific to case where WBIC is used for landing control because, unlike the centroidal dynamics model using by the AAKD planner, the lumped-mass model used in MPC does not agree with the WBIC model in a straightforward way.

One issue is defining the orientation task. In the MPC, the orientation is simply defined by the orientation of the lumped mass. However, in the case of the humanoid robot, the body orientation has a different meaning than the lumped mass orientation because the body does not include all other link inertias. The most reasonable choice is the orientation of centroidal inertia [17], however, controlling the angle of eigenvectors of the inertia does not exactly coincide with the output of MPC and difficult to add as a WBIC task. Fortunately, the mass of MIT Humanoid is concentrated around its body, and the actuators that dominante to robot's mass do not change location or orientation much while the robot moves. Therefore, we can approximate that the body orientation represents the orientation of the lumped mass and use that orientation as the reference of the body orientation task in WBIC.

Another issue is selecting the task priorities. More specifically, the place of centroidal momentum (CM) task and how to define the task are the main question. The linear portion of CM is straightforward since it is simply CoM motion control, but the angular part is not easy as we partially described above. Minimizing CAM is desirable in most cases but not always since sometimes a robot must rotate its body to ensure the overall balance in the long-term perspective. Therefore, we take the orientation reference computed in MPC as a command of the body orientation as we mentioned above. The body orientation task is selected as the first task in the task prioritization. We locate the CM task under this task. The feedback law for the CM task does not include orientation error correction, but only centroidal angular velocity tracking. By prioritizing the tasks like this, we track the full-body angular momentum as long as the robot's orientation follows the reference input.

To formulate null space projection of CM task, we need to define the Jacobian of the CM. [28] suggested the Jacobian for CM inertia shaping, but in this work, we care more about CM motion than the shape of its inertia. Therefore, we use the Jacobian defined in [29], $J_{\text{CM}} = I_{\text{CM}}^{-1} A_{CM}$. The other tasks below the CM task such as the joint posture task recognize the CAM minimization intention since the tasks are projected onto the null-space of the CM task, and automatically find an arm motion to counteract the body orientation control, which you can see in our resultant motions.

IV. RESULTS

The simulation environment we use for testing the proposed system design is the custom dynamics simulator designed by the MIT Biomimetic Robotics Lab to address actuator dynamics as well as multi-DoF rigid-body dynamics of the robot. The performance of the simulator is extensively tested in our previous work using Mini-Cheetah robot, and the same framework for simulation actuator and multi-body dynamics is used for humanoid simulation experiments. Specifically, the model of the simulated actuators includes the effect of rotor inertias on the robot's dynamics, the torque-speed relationships of the motors, and the effect of battery voltage droop.

The dynamic capabilities of the humanoid robot are showcased through demonstrations of acrobatic abilities such as aerial flips and spins. While these acrobatic motions are not explicitly useful for legged robots, they serve as an important benchmark for what behaviors will be possible on the robot as future controllers are developed (e.g. parkour). All motion plans are generated using the AAKD planner discussed in Section III. The combination of terminal constraints and consideration of the robot's actuator limits allows trajectories to be rapidly generated and ready for immediate deployment on the robot. A comparison of the optimized joint torque profiles for standard kino-dynamic (KD) planning of a 180° spinning jump versus AAKD planning for the same jump shown in Fig. 7. This jumps exhibits relatively large displacements of the hip yaw and ab/ad joints, making it a worst-case scenario for the contact Jacobian approximation (14). Despite this, all actuator limits are respected in the AAKD case, although they are not respected in the KD case. The results in Table III compare the solve times for various acrobatic motions and show that AAKD only increasing solve times with respect to KD by an average of 28.2%s
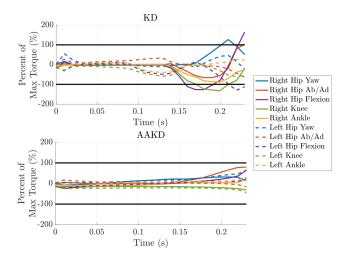
Fig. 6. **Actuation-Aware Planning.** Optimized joint torque profiles for a standing front flip of the MIT Humanoid planned with (top) a standard kino-dynamic planner (top) and (bottom) the proposed AAKD planner. Note that these profiles are only for the takeoff portion of the jump, landing and flight are not included.

TABLE III
SOLVE TIMES FOR KINO-DYNAMIC PLANNERS

	180° Spin	Forward Jump	Lateral Jump	Front Flip	Back Flip
KD	4.09s	2.43s	2.98s	3.35s	6.51s
AAKD	4.67s	3.10s	3.24s	5.29s	8.77s

The AAKD planner produces motion that can coordinate high speed vertical propulsion of the robot's CoM with large angular momentum generation about the robot's principal axes to create the acrobatic motions demonstrated in this work, a few of which are shown in Fig. 7. Notice in these sequential screenshots that arm motions are used to stabilize the posture of the body. These arm motions are not authored by the AAKD planner, but rather, they emerge from the centroidal momentum task of the WBIC.

The high bandwidth full-body control offered by the WBIC makes it a useful for both the takeoff and landing phases of the demonstrated dynamic motions. The WBIC is able to account for the errors in the kino-dynamic and lumped-mass models of the robot as well as recover from small perturbations to nominal trajectories. An example of the tracking performance of the WBIC during the takeoff phase of the front flip is shown in Fig. 8. The front flip is focused on in the discussion of the results because it involves the most significant rotation and the most severe impact with the ground, making it the worst-case scenario for both the takeoff and landing portions of the control framework. All other dynamic motions demonstrated in the attached video likewise exhibit robust tracking and obey all actuator limits.

While the acrobatic motions presented in this work are strictly simulation results, we again emphasize that extraordinary care is taken to ensure that the simulated dynamics reflect the true dynamics of the robot hardware. First, we include the inertia of each of the rotor inertia and account for the non-trivial effects that these spinning rotors have on
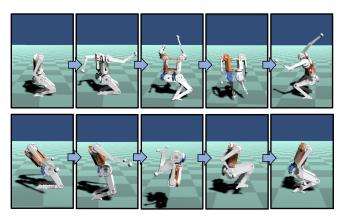

Fig. 7. **Acrobatic Humanoid.** MIT Humanoid performing a 180° spinning jump (top) and a standing front flip (bottom)
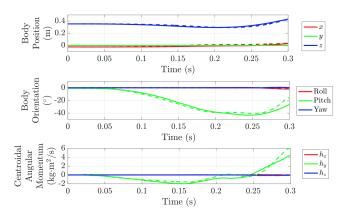

Fig. 8. **WBIC for Takeoff.** WBC tracking performance for the takeoff portion of the standing front flip of the MIT Humanoid.

the dynamics of the robot. We enforce that the simulated actuators obey the empirically verified torque-speed relationships illustrated in Fig. 4. Lastly, we verify that the effects of battery voltage droop are reflected in the torque limits of the robot. The joint torque data in Fig. 9 shows that the simulated torques exerted by the robot throughout all phases of motion virtually never exceed the limits imposed by our empirically verified actuator model. In these results, the experimental limit does not account for the effect of field weakening, so in reality the limit will be higher in many areas.

Since the actuators have been modeled great detail, only the following assumptions about the simulated system remain. First, we assume our model is accurate in terms of inertial properties. Second, we assume our rigid contact model of the ground is suitably accurate. Lastly, we assume the robot can adequately estimate its state. If these assumptions hold, which they likely will based on previous experiments with the MIT Cheetah robots, then we can be nearly certain that all demonstrations shown in this paper are ready for hardware implementation.

V. CONCLUSIONS

In this paper, we present a system design for the acrobatic motion demonstration of a humanoid robot by proposing a new humanoid robot, motion planner, and landing controller.
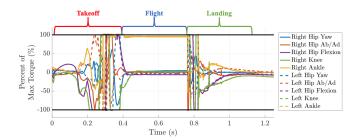
Fig. 9. **Simulated Actuator Model.** Simulated torque profiles of the MIT Humanoid performing a front flip.

Two new actuators are developed to satisfy the torque and power requirement for impulsive behaviors. On the software side, we formulate a new torque limit constraint that can be integrated into the simplified model based kino-dynamic planner to find a feasible trajectory in a real robot. The optimal trajectories are simulated in our custom dynamics simulator including actuator models and the resultant actuator outputs are verified again with a complete torque-velocity actuator model including battery voltage droop and back-EMF effect. For landing, we integrate MPC and WBIC with special care for dynamical consistency in transferring the optimal output of MPC to WBIC orientation control.

ACKNOWLEDGMENTS

REFERENCES

[1] B. Dynamics, "What's new, atlas?" https://youtu.be/fRj34o4hN4I, 2017, accessed on 2021/01/21.

[2] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous robots*, vol. 40, no. 3, pp. 429–455, 2016.

[3] K. Kaneko, M. Morisawa, S. Kajita, S. Nakaoka, T. Sakaguchi, R. Cisneros, and F. Kanehiro, "Humanoid robot hrp-2kai—improvement of hrp-2 towards disaster response tasks," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 132–139.

[4] T. Jung, J. Lim, H. Bae, K. K. Lee, H.-M. Joe, and J.-H. Oh, "Development of the humanoid disaster response platform drc-hubo+," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 1–17, 2018.

[5] H. Dai and R. Tedrake, "Planning robust walking motion on uneven terrain via convex optimization," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 579–586.

[6] R. J. Griffin, G. Wiedebach, S. McCrory, S. Bertrand, I. Lee, and J. Pratt, "Footstep planning for autonomous walking over rough terrain," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2019, pp. 9–16.

[7] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, "Optimization-based full body control for the darpa robotics challenge," *Journal of Field Robotics*, vol. 32, no. 2, pp. 293–312, 2015.

[8] G. Nelson, A. Saunders, N. Neville, B. Swilling, J. Bondaryk, D. Billings, C. Lee, R. Playter, and M. Raibert, "Petman: A humanoid robot for testing chemical protective clothing," *Journal of the Robotics Society of Japan*, vol. 30, no. 4, pp. 372–377, 2012.

[9] S.-H. Hyon, D. Suewaka, Y. Torii, and N. Oku, "Design and experimental evaluation of a fast torque-controlled hydraulic humanoid robot," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 2, pp. 623–634, 2016.

[10] N. A. Radford, P. Strawser, K. Hambuchen, J. S. Mehling, W. K. Verdeyen, A. S. Donnan, J. Holley, J. Sanchez, V. Nguyen, L. Bridgwater *et al.*, "Valkyrie: Nasa's first bipedal humanoid robot," *Journal of Field Robotics*, vol. 32, no. 3, pp. 397–419, 2015.

[11] N. G. Tsagarakis, D. G. Caldwell, F. Negrello, W. Choi, L. Baccelliere, V.-G. Loc, J. Noorden, L. Muratore, A. Margan, A. Cardellino *et al.*, "Walk-man: A high-performance humanoid platform for realistic environments," *Journal of Field Robotics*, vol. 34, no. 7, pp. 1225–1259, 2017.

[12] P. M. Wensing, A. Wang, S. Seok, D. Otten, J. Lang, and S. Kim, "Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots," *Ieee transactions on robotics*, vol. 33, no. 3, pp. 509–522, 2017.

[13] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.

[14] I. Mordatch, K. Lowrey, and E. Todorov, "Ensemble-cio: Full-body dynamic motion planning that transfers to physical humanoids," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 5307–5314.

[15] P. M. Wensing and D. E. Orin, "High-speed humanoid running through control with a 3d-slip model," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 5134–5140.

[16] A. Hereid, M. J. Powell, and A. D. Ames, "Embedding of slip dynamics on underactuated bipedal robots through multi-objective quadratic program based control," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 2950–2957.

[17] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous robots*, vol. 35, no. 2-3, pp. 161–176, 2013.

[18] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 295–302.

[19] A. Herzog, S. Schaal, and L. Righetti, "Structured contact force optimization for kino-dynamic motion generation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2703–2710.

[20] D. Kim, J. D. Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," 2019.

[21] S. Seok, A. Wang, M. Y. Chuah, D. Otten, J. Lang, and S. Kim, "Design principles for highly efficient quadrupeds and implementation on the mit cheetah robot," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3307–3312.

[22] H.-W. Park, S. Park, and S. Kim, "Variable-speed quadrupedal bounding using impulse planning: Untethered high-speed 3d running of mit cheetah 2," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5163–5170.

[23] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2245–2252.

[24] B. Katz, J. Di Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6295–6301.

[25] ——, "Mini Cheetah - A Platform for Pushing the Limits of Dynamic Quadruped Control." in *International Conference on Robotics and Automation (ICRA)*, 2019.

[26] P. M. Wensing and D. E. Orin, "Improved computation of the humanoid centroidal dynamics and application for whole-body control," *International Journal of Humanoid Robotics*, vol. 13, no. 01, p. 1550039, 2016.

[27] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.

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

[29] D. Kim, S. J. Jorgensen, P. Stone, and L. Sentis, "Dynamic behaviors on the nao robot with closed-loop whole body operational space control," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 1121–1128.