The Surprising Effectiveness of Linear Models for Whole-Body Model-Predictive Control

Arun L. Bishop[1], Juan Alvarez-Padilla[1], Sam Schoedel[1], Ibrahima Sory Sow[1], Juee Chandrachud[1],
Sheitej Sharma[1], Will Kraus[1], Beomyeong Park[2], Robert J. Griffin[2], John M. Dolan[1], Zachary Manchester[1]

Fig. 1: The hydraulic humanoid Nadia walking forward over multiple steps using our convex MPC formulation with linear time-invariant dynamics constraints. The gait has a stride length of 30 cm, a swing phase of 0.6 s and a double support phase of 0.3 s.

Abstract— **When do locomotion controllers require reasoning about nonlinearities? In this work, we show that a whole-body model-predictive controller using a simple linear time-invariant approximation of the whole-body dynamics is able to execute basic locomotion tasks on complex legged robots. The formulation requires no online nonlinear dynamics evaluations or matrix inversions. We demonstrate walking, disturbance rejection, and even navigation to a goal position without a separate footstep planner on a quadrupedal robot. In addition, we demonstrate dynamic walking on a hydraulic humanoid, a robot with significant limb inertia, complex actuator dynamics, and large sim-to-real gap.**

I. INTRODUCTION

Legged locomotion is a key capability for robots navigating human spaces. These robots are usually underactuated and have non-smooth and nonlinear dynamics, posing a challenging planning and control problem that has been the focus of significant work over the last several decades. Recent methods have achieved impressive results by reasoning about the full nonlinear dynamics of the robot and the corresponding non-convex optimal control problem, either online using nonlinear model-predictive control (NMPC) or offline using learning-based methods such as reinforcement learning (RL). However, these methods are computationally expensive, difficult to tune, and hard to analyze, leading to an open question: what level of reasoning about these complex nonlinear dynamics is required for locomotion? In this paper we demonstrate that basic locomotion tasks are achievable using a single time-invariant linear dynamics model and convex model-predictive control (MPC).

Some of the earliest legged locomotion work hinted that these challenging systems have significant underlying structure. By leveraging simple models such as the spring-loaded inverted pendulum (SLIP) [1] along with the zero-moment point [2][3] and support polygon, researchers were able to construct some of the first successful locomotion controllers for walking and running. These approaches resulted in simple and often closed-form control laws that lent themselves to analysis of their stability and robustness properties [4][5][6], and yielded surprising results, such as a blind open-loop controller for stable running over rough terrain [7]. While critical to developing our understanding of locomotion fundamentals, these methods have been challenging to generalize, and have limited ability to reason about leg inertias, constraints, and disturbances.

To address these limitations, researchers have turned to optimization-based methods. One widely used approach is MPC, which formulates and solves a receding-horizon optimal control problem given dynamics, constraints, and a cost function. These problems have classically been solved using second-order (e.g. Newton-type) methods, [8][9][10] though more recently sampling-based methods [11] have also achieved promising results. MPC approaches have demon-

[1] Authors are with the Robotics Institute, the Department of Electrical and Computer Engineering, and the Department of Mechanical Engineering, Carnegie Mellon University
[2] Authors are with the Florida Institute for Human and Machine Cognition

strated agile and robust behaviors, perhaps most famously demonstrated by Boston Dynamics [12][13]. However, a primary challenge of MPC approaches has been solving large optimization problems fast enough online.

A common approach to solve MPC problems online is sequential quadratic programming (SQP). In SQP, the dynamics and constraints are linearized along the previous solution iterate and a quadratic approximation of the cost function is used to form a quadratic program (QP) [14]. The solution to this subproblem is then used to compute a new iterate. Computing a solution to these QPs involves solving a potentially large linear system, which is computationally expensive and challenging to parallelize [15][16]. In addition, while each subproblem may be convex, the general nonlinear MPC problem is often non-convex, with no convergence guarantees. Significant domain knowledge and systems engineering is required to achieve good performance in practice [17].

Given that constructing and solving a QP at each iteration is a significant bottleneck, a natural question to ask is: how much information can we leverage from each subproblem? For example, when running MPC online, the subproblems often don't change significantly between solves if the dynamics are much slower than the control rate. The resulting QP also has an equivalent closed-form linear feedback policy if the active constraint set doesn't change. Some nonlinear MPC approaches have leveraged these facts to reduce the effect of solver delay [18][19]. In addition, many works have shown the power of LQR controllers [20] and linearized dynamics models for complex problems like airplane perching [21]. Beyond just the QP setting, linear feedback policies have been shown to be surprisingly capable. For example, [22] [23] showed that learned linear policies can stabilize gait generators for walking on rough terrain.

In this work, we explore the efficacy of linear models by probing the capability of an MPC controller using a single time-invariant linearized whole-body dynamics model of a legged robot. We hypothesize that this convex MPC formulation is sufficient for many practical locomotion tasks that do not require extreme joint angles or body orientations. Our formulation is:

- **Convex**, yielding a unique solution in polynomial time;
- **Linear**, with no online nonlinear dynamics or kinematics evaluations;
- **Matrix-inverse-free**, requiring only matrix-vector products and clamp operations online.

First, we demonstrate our controller on a range of tasks on a Unitree Go2 quadruped, showing that it is able to recover from perturbations and walk to a goal position with up to 90 degrees of yaw error when given only a walking-in-place reference with no additional higher-level planner. We also test stepping onto a 24 cm box and show the controller is able to handle the change in pitch and foot height.

Second, we demonstrate our controller on a hydraulic humanoid, a more challenging system due to reduced control authority, patch contacts, large leg inertias, and a significant sim-to-real gap. We show that it can stably walk in-place

and is able to walk forward with a speed of 0.17 m/s and a stride length of 0.3 m, with a swing phase of 0.6 s and a double support phase of 0.3 s, which cannot be performed quasi-statically.

II. BACKGROUND

A. Dynamics Model

We model each legged robot using a floating base, where the configuration q consists of the base position in world coordinates, a body-to-world quaternion, and the joint angles. The velocity vector v consists of the linear and angular velocities of the base in the body frame and the joint velocities. We can relate $\dot{q}$ to v using the following velocity kinematics, where $E(q)$ contains the attitude Jacobian for the quaternion [24].

$$\dot{q} = E(q)v. \tag{1}$$

The continuous-time manipulator equations for a robot with external forces is

$$M(q)\dot{v} + C(q, v) = B\tau + J(q)^T \lambda, \tag{2}$$

where $M(q)$ is the mass matrix, $C(q, v)$ is the dynamics bias, B is the input Jacobian mapping actuator torques to generalized forces and τ are motor torques. $\lambda \in \mathbb{R}^{3N_c}$ is a vector of contact forces in the world frame (x, y, z) for each contact point where N_c is the number of contact points, and $J(q)$ is a Jacobian that maps the external contact forces to generalized forces, which is described in more detail in Section III-C. These dynamics are put in state-space form and discretized using an integrator such as backward Euler or Runge-Kutta.

B. Contact Modeling

Contact dynamics are classically modeled as complementarity constraints, which encode that either the distance between the objects in contact or the contact force is zero. Adding that both the distance and contact force must be non-negative results in the following three constraints.

$$\phi(q) \geq 0, \tag{3}$$
$$n \geq 0, \tag{4}$$
$$\phi(q)n = 0, \tag{5}$$

where $\phi(q)$ is the signed distance as a function of configuration and n is the normal force.

Solving the general set of constraints above results in a problem that is non-convex and can be ill-conditioned or have singular Jacobians. Instead, we adopt the hybrid approach to model these constraints in our MPC problem: We pre-specify whether objects are in contact or not (referred to as the contact mode) at each time step. We can then encode contact directly through equality and inequality constraints:

$$\phi(q) \geq 0, \quad n = 0 \qquad \text{(no contact)}$$
or
$$\phi(q) = 0, \quad n \geq 0 \qquad \text{(contact)}$$

By dropping the complementarity constraints and fixing the contact modes, we can convexify the constraints. However, the controller can no longer plan over the ordering of contact events.

C. Trajectory Optimization and Nonlinear MPC

Trajectory optimization and nonlinear MPC problems can be formulated as:

$$\min_{x_{1:N}, u_{1:N-1}} \sum_{k=1}^{N-1} \ell_k(x_k, u_k) + \ell_f(x_n) \quad (6)$$

$$\text{s.t.} \quad x_1 = x_{i.c.}$$
$$f(x_k, u_k, x_{k+1}) = 0 \quad \forall k \in 1, \dots, N-1$$
$$x_k \in \mathcal{X}, u_k \in \mathcal{U} \quad \forall k \in 1, \dots, N-1$$

In this problem $x_k = [q_k; v_k]$ is the state at time k, u_k is the control, ℓ_k is the stage cost, ℓ_f is the terminal cost, f is a discrete dynamics constraint, and $\mathcal{X}$ and $\mathcal{U}$ are sets representing general state and control constraints. These problems are typically nonlinear and nonconvex, with no guarantee of finding a globally optimal solution.

III. CONTROLLER FORMULATION

In this section, we describe the specific dynamics, constraints, and costs of our linear time-invariant MPC formulation, which results in a quadratic program that is solved online.

A. Low-Level Control

Many robots, including the Unitree Go2, include low-level PD controllers in their motors that operate with much lower latency than is possible through higher-level torque-control interfaces. While our methodology is capable of fast whole-body torque control, we find that the additional latency introduced by the Go2 torque-control interface limits performance. Therefore, we modify the dynamics (2) to include these low-level PD controller dynamics,

$$M\dot{v} + C + K_p q + K_d \dot{q} = B\tau, \quad (7)$$
$$\tau = \tau_{ff} + B^T(K_p q_d + K_d \dot{q}_d),$$

where K_p and K_d are diagonal gain matrices and τ_{ff} is a feed-forward torque command.

B. Dynamics Linearization

When linearizing the floating-base state, we use an axis-angle rotation parameterization and define the state error $\Delta x = e(x, x_0)$, where we use quaternion operations to compute the attitude error and vector subtraction otherwise. We combine Eq. 1 and Eq. 2 into a continuous-time state-space model and linearize them about x_0, u_0, λ_0 and then apply backward-Euler integration to construct the following affine linear dynamics model.

$$A^+ \Delta x_{k+1} + A\Delta x_k + B\Delta u_k = d_k \quad (8)$$
$$\text{where } \Delta u_k = [\Delta \tau_k; \; \Delta \lambda_k]$$

A problem we leave to future work is the best choice of linearization point. Intuitively, we desire one that minimizes the prediction error over the operating space of interest. For our locomotion tasks, we chose to use a standing pose and the corresponding controls and contact forces for x_0, u_0, and λ_0.

C. Locomotion Contact Constraints

Given a single contact point, let $c(q) : \mathbb{R}^{nq} \to \mathbb{R}^3$ map the configuration of the robot to the world coordinates of the contact point and let $J(q) = \frac{\partial c}{\partial q}$. Our constraint restricts the x and y positions at adjacent knot points to be equal when in contact, letting the MPC controller choose where to place the contact. Assuming that the world frame has the contact height at z = 0, we have the following constraint for each contact point.

$$c(q_{k+1}) - \text{diag}(\begin{bmatrix} 1 & 1 & 0 \end{bmatrix})c(q_k) = 0 \quad (9)$$

The resulting linearized constraint is then as follows.

$$J\Delta q_{k+1} - \text{diag}(\begin{bmatrix} 1 & 1 & 0 \end{bmatrix})J\Delta q_k = 0. \quad (10)$$

When not in contact, the constraint bounds are set to a lower bound of $(-\infty, -\infty, 0)$ and an upper bound of (∞, ∞, ∞).

D. Reference Tracking

We provide whole-body gait references of the state, control, and contact forces for the controller to track, which we represent as $\hat{x}$ and $\hat{u}$. Since we have a single linearization point, the MPC dynamics plan over $\Delta x = e(x, x_0)$ and $\Delta u = u - u_0$. Similarly, we define the reference with respect to the linearization point as $\Delta \hat{x}$ and $\Delta \hat{u}$. For the rest of this section we'll drop the Δ for brevity and assume x, u, $\hat{x}$ and $\hat{u}$ are with respect to x_0 and u_0.

$$\min_{x_{1:N}, u_{1:N-1}, \lambda_{1:N-1}} \sum_{k=1}^{N-1} \frac{1}{2} \left[(x_k - \hat{x}_k)^T Q(x_k - \hat{x}_k) + (u_k - \hat{u}_k)^T R(u_k - \hat{u}_k) \right] + \frac{1}{2}(x_N - \hat{x}_N)^T Q_f(x_N - \hat{x}_N) \quad (11)$$

$$\text{s.t.} \quad x_1 = x_{i.c.} \quad \text{(initial state)}$$
$$A^+ x_{k+1} + Ax_k + Bu_k = d_k \quad \text{(dynamics)}$$
$$l_c \leq Jx_{k+1} \leq h_c \quad \text{(contact mode constraints)}$$
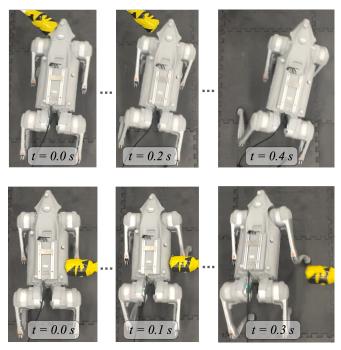$$l_f \leq u_k \leq h_f \quad \text{(force and torque constraints)}$$

Fig. 2: Top-down view of two perturbations applied to the Unitree Go2 while stepping in place. The robot maintains contact with the ground without slipping and return to its original location.

E. Linear Time-Invariant MPC

The final formulation for the linear time-invariant MPC controller is shown in (11). It consists of a quadratic tracking cost, an initial condition constraint, and three constraints at each time step: a dynamics constraint, a contact mode constraint, and a friction-cone force constraint. This problem is a quadratic program and can be written in the following general form,
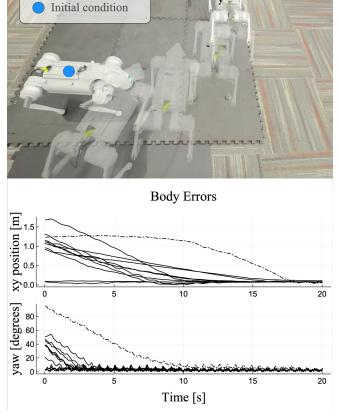
$$\min_{z} \frac{1}{2} z^T H z + g^T z \tag{12}$$

$$\text{s.t. } l \le Dx \le h, \tag{13}$$

since the constraint Jacobian D and cost Hessian H are fixed. This formulation allows certain QP solvers, such as ReLU-QP and OSQP, to pre-factorize or pre-invert the KKT system offline (which is a $\mathcal{O}(n^3)$ operation) and then perform only back-substitutions or matrix-vector products online to solve the QP (which are $\mathcal{O}(n^2)$ operations). The reference and contact modes can be changed by only modifying the cost gradient g and/or constraint bounds l and h, which are shown in dark red in (11).

IV. EXPERIMENTS

We tested our MPC formulation on two systems: a quadruped and a humanoid. Feasible reference motions were planned offline using dynamics from Pinocchio and the trajectory optimization library Aligator [25], and the QP was solved online using OSQP [26]. Videos and code can be found on our project website `linearwalking.github.io`.

Fig. 3: A Unitree Go2 robot starts from 10 initial conditions with yaw from 5 to 90° and displacements from 0.1 to 1.7 m. Transparent overlays show keyframes starting with 90° yaw and 1.3 m displacement. Our controller is able to return to the initial position without assistance from a footstep planner.

A. Quadruped Experiments

For our quadruped experiments, we used the Unitree Go2, which has 12 joints (three per leg). It is relatively lightweight, with a total mass of 15.7 kg. While each leg weighs about 2 kg, the mass is centered near the hip, resulting in low angular inertias. We modeled each foot as a single contact point and used constraints on both position and velocity, resulting in 24 constraints and 24 contact forces. The resulting linearized dynamics model has 36 states, 12 actuator torques, and 24 contact forces.

We used a planning horizon of 0.2 seconds with a timestep of 0.01 seconds. The closed-loop control rate was approximately 500 Hz. Experiments were run on a workstation computer equipped with an Intel i9-12900KS CPU and 64 GB of RAM. We performed three experiments: disturbance rejection, goal navigation, and stepping onto a box. For the first two, we used a walking-in-place reference generated with Aligator. No additional online high-level planner or contact heuristics were used during the experiments — all behaviors came purely from the MPC controller.

1) Disturbance Rejection: We tested disturbance rejection by pushing the robot from the front and side while it stepped in place. Fig. 2 shows selected frames from the experiments
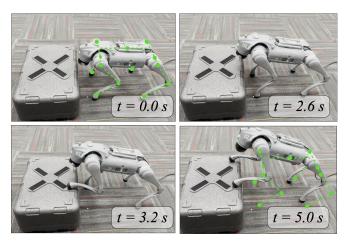
Fig. 4: Keyframes of a Unitree Go2 robot stepping onto a 24 cm box. The robot tracks a trajectory with up to 60 degrees joint deviation and 10 degrees pitch deviation from the linearization pose, with the green dashed skeleton showing the linearization pose.

demonstrating that the controller is capable of recovering from moderate pushes and exhibits footstep recovery behaviors despite relying on a single linearization. However, we found that large or sustained perturbations resulted in the controller failing due to an infeasible QP where the contact constraints were not satisfiable. This problem could be alleviated with a footstep planner.

2) Goal Navigation: Using an open-loop walking-in-place reference, we demonstrate that the linear controller is able to plan footsteps to navigate to a goal position despite large changes in orientation. The top of Fig. 3 shows the quadruped starting away from the origin with a yaw error of 90 degrees and walking to the origin over 18 seconds, driving the body position and orientation errors to zero. To further validate our controller, we initialized the quadruped from 10 different poses that span yaw errors from five to 90 degrees and 0.1 m to 1.7 m body displacements from the linearization pose. The bottom of Fig. 3 illustrates the convergence of both the position and orientation errors to zero across all trials, where the dash-dot trajectories correspond with the frames shown in the top figure.

3) Stepping onto a Box: We demonstrate that the controller is able to handle large changes in pitch and footstep height. Fig. 4 shows four keyframes from the robot stepping onto a box with a height of 24 cm, resulting in maximum deviations of 10 degrees in the body pitch, 11 degrees in the hip joints, 29 degrees in the thigh joints, and 60 degrees in the calf joints from the linearization pose. Handling a different contact height is simple, and involves updating the contact constraint from $J_z \Delta x = 0$ to $J_z \Delta x = J_z \Delta \hat{x}$ where $\hat{x}$ is the reference.

B. Hydraulic Humanoid

To further test our method, we performed a set of walking experiments on IHMC's Nadia, a hydraulic humanoid. This problem is more challenging for several reasons: Unlike the quadruped, Nadia has hydraulic linear actuators that drive the leg joints through linkage mechanisms. Each joint is approximated as a revolute joint with torque actuators and converted to hydraulic forces by a low-level control stack. While this approximation is exact for most joints, the knee has a four-bar linkage where the revolute joint approximation is inaccurate. Each leg weighs 21 kg, approximately 23% of the total mass, and has the center of mass close to the knee. In addition, the feet form patch contacts with the ground and the support polygon and capture regions are much smaller [2][3].

We model each foot with four contact points at its corners. This makes the normal force constraint linear, but the forces are no longer unique and the contact Jacobians are rank deficient. We resolve this issue by adding a small regularizer (10^{-5}) to the contact forces constraints.

For our MPC problem, we chose to focus on control of the lower body, since we had limited access to the hardware. This leaves 12 revolute joints, six for each leg, and one yaw joint at the spine. The resulting linearized dynamics model has 38 states, 13 actuator torques, and 24 contact forces. We planned over a 0.2 second horizon with a timestep of 0.01 seconds, and ran 20 QP iterations for each solve. The closed loop control rate was approximately 333 Hz, which was based off the frequency that the robot's built-in control stack runs at. Experiments were run on a workstation computer equipped with an Intel i7-13700KF CPU.

To approximately quantify the speed gain achieved by using a fixed Hessian and constraint Jacobian and avoiding matrix factorizations online, we measured the factorization and backsolve timing for QDLDL, OSQP's linear system solver, on Nadia's KKT system which is a 3,504 x 3,504 matrix that is 99.3% sparse. QDLDL takes $5.9 \pm 0.8 ms$ to factorize the KKT system but only $0.10 \pm 0.02 ms$ to perform a back-solve. This enables our single convex MPC controller to run significantly faster than an NMPC approach, which would additionally need to evaluate nonlinear dynamics, compute Jacobians, and possibly employ additional steps such as a line-search or outer penalty loops.

We demonstrate our MPC controller on hardware for three different reference motions: walking in-place, and walking forward with short (17 cm) and long (30 cm) stride lengths. The parameters for each reference are shown in Table I. We found the robot was able to stably walk in place, walking for over a minute with no noticeable instabilities. We also ran six trials for both forward walking references consisting of eight steps each with no failures. Figure 1 shows each step from one of the trials on hardware. Figure 5 shows the foot locations and center of mass trajectories in the XY plane for all 6 runs with a stride length of 30 cm, with the reference foot locations and center of mass trajectory shown with dashed lines. The reference trajectory is not quasi-static, as shown by the center of mass staying out of the support polygon of the stance foot for each step. The maximum final tracking error for the center of mass across all six runs was 3.6 cm.
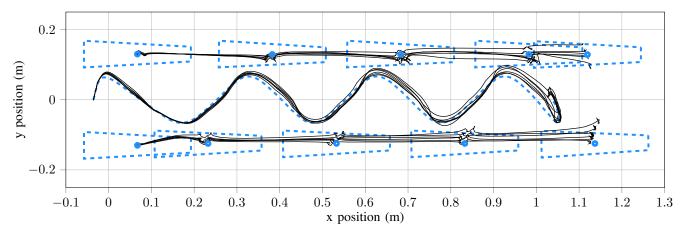
Fig. 5: Foot and center-of-mass trajectories across six humanoid walking trials (black) over the desired foot placement positions and center of mass trajectory (blue, dashed). Our controller is able to track the trajectory with minor deviations despite large modeling errors.

Reference	Stride	Single Support	Double Support
In-place	NA	0.8 s	0.6 s
Forward short	17 cm	0.8 s	0.6 s
Forward long	30 cm	0.6 s	0.3 s

TABLE I: Gait parameters for the three humanoid walking behaviors demonstrated on hardware.

V. DISCUSSION AND CONCLUSIONS

We have presented a linear time-invariant MPC controller that is convex and avoids both online nonlinear dynamics evaluations and costly matrix factorizations. We demonstrated the resulting controller on a variety of quadruped tasks to show its ability to handle orientation deviations and reason about foot placement. In addition, we showed that our controller can perform walking on a hydraulic humanoid with large leg inertias and complex kinematics.

A. Limitations

We have focused on a simple formulation — avoiding a higher-level planner and nonlinear evaluations — to isolate the capabilities of our single-linearization formulation. However, this does result in some clear limitations: Without contact re-planning, large disturbances can lead to infeasible QPs and controller failure, which we observed during large-perturbation experiments. We also found that during bipedal walking the controller struggled to keep the robot's feet flat. In addition, the cost Hessian is also fixed in our approach, preventing task-specific cost tuning, for example for swing-leg tracking.

B. Future Work

There are many potential directions for future work: First, some of the issues noted above could be addressed with a learned or heuristic-based planner that could update contact modes, preventing infeasible QPs. Second, thanks to its low computational cost, our controller could be embedded directly into the training pipeline of an RL policy reasoning about constraints. Third, some nonlinear dynamics evaluations could also be incorporated to increase performance with very minimal additional computational cost. Lastly, switching between multiple linear models online could enable highly dynamic motions with very large joint-angle or body attitude deviations.

VI. ACKNOWLEDGMENTS

REFERENCES

[1] R. Blickhan, "The spring-mass model for running and hopping," eng, *Journal of Biomechanics*, vol. 22, no. 11-12, pp. 1217–1227, 1989.

[2] M. Vukobratović and J. Stepanenko, "On the stability of anthropomorphic systems," *Mathematical Biosciences*, vol. 15, no. 1, pp. 1–37, Oct. 1972.

[3] M. Vukobratović and B. Borovac, "Zero-Moment Point — Thirty Five Years of its Life," en, *International Journal of Humanoid Robotics*, vol. 01, no. 01, pp. 157–173, Mar. 2004.

[4] M. Ernst, H. Geyer, and R. Blickhan, "Spring-Legged Locomotion on Uneven Ground: A Control Approach to Keep the Running Speed Constant," en, in *Mobile Robotics*, Istanbul, Turkey: WORLD SCIENTIFIC, Aug. 2009, pp. 639–644.

[5] M. Buhler and D. Koditschek, "Analysis of a simplified hopping robot," in *1988 IEEE International Conference on Robotics and Automation Proceedings*, Apr. 1988, 817–819 vol.2.

[6] M. Spong and G. Bhatia, "Further results on control of the compass gait biped," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 2, Oct. 2003, 1933–1938 vol.2.

[7] R. Altendorfer, D. E. Koditschek, and P. Holmes, "Stability Analysis of a Clock-Driven Rigid-Body SLIP Model for RHex," *The International Journal of Robotics Research*, vol. 23, no. 10-11, pp. 1001–1012, Oct. 2004, Publisher: SAGE Publications Ltd STM.

[8] G. Bledt, P. M. Wensing, and S. Kim, "Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the MIT cheetah," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, ISSN: 2153-0866, Sep. 2017, pp. 4102–4109.

[9] J. Koenemann et al., "Whole-body model-predictive control applied to the HRP-2 humanoid," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 3346–3351.

[10] M. Neunert et al., "Whole-Body Nonlinear Model Predictive Control Through Contacts for Quadrupeds," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1458–1465, Jul. 2018.

[11] J. Alvarez-Padilla, J. Z. Zhang, S. Kwok, J. M. Dolan, and Z. Manchester, "Real-Time Whole-Body Control of Legged Robots with Model-Predictive Path Integral Control," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, arXiv:2409.10469 [cs], Sep. 2024.

[12] Robotics Today, *"Recent Progress on Atlas, the World's Most Dynamic Humanoid Robot" - Scott Kuindersma*, Jun. 2020.

[13] Boston Dynamics, *Atlas — Partners in Parkour*, Aug. 2021.

[14] J. Nocedal and S. J. Wright, *Numerical Optimization* (Springer Series in Operations Research and Financial Engineering), en. Springer New York, 2006.

[15] M. Schubiger, G. Banjac, and J. Lygeros, "GPU acceleration of ADMM for large-scale quadratic programming," *Journal of Parallel and Distributed Computing*, vol. 144, pp. 55–67, Oct. 2020.

[16] A. L. Bishop, J. Z. Zhang, S. Gurumurthy, K. Tracy, and Z. Manchester, "ReLU-QP: A GPU-Accelerated Quadratic Programming Solver for Model-Predictive Control," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, May 2024, pp. 13 285–13 292.

[17] E. F. Camacho and C. Bordons, "Nonlinear Model Predictive Control: An Introductory Review," en, in *Assessment and Future Directions of Nonlinear Model Predictive Control*, R. Findeisen, F. Allgöwer, and L. T. Biegler, Eds., Berlin, Heidelberg: Springer, 2007, pp. 1–16.

[18] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, "Feedback MPC for Torque-Controlled Legged Robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, ISSN: 2153-0866, Nov. 2019, pp. 4730–4737.

[19] H. Li and P. M. Wensing, "Cafe-Mpc: A Cascaded-Fidelity Model Predictive Control Framework With Tuning-Free Whole-Body Control," *IEEE Transactions on Robotics*, vol. 41, pp. 837–856, 2025.

[20] R. Tedrake, S. Kuindersma, R. Deits, and K. Miura, "A closed-form solution for real-time ZMP gait generation and feedback stabilization," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, Nov. 2015, pp. 936–940.

[21] J. Moore and R. Tedrake, "Control synthesis and verification for a perching UAV using LQR-Trees," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, ISSN: 0743-1546, Dec. 2012, pp. 3707–3714.

[22] M. Rahme, I. Abraham, M. L. Elwin, and T. D. Murphey, "Linear Policies are Sufficient to Enable Low-Cost Quadrupedal Robots to Traverse Rough Terrain," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, ISSN: 2153-0866, Sep. 2021, pp. 8469–8476.

[23] L. Krishna, G. A. Castillo, U. A. Mishra, A. Hereid, and S. Kolathaya, *Linear Policies are Sufficient to Realize Robust Bipedal Walking on Challenging Terrains*, arXiv:2109.12665 [cs], Oct. 2021.

[24] B. E. Jackson, K. Tracy, and Z. Manchester, "Planning With Attitude," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5658–5664, Jul. 2021, Conference Name: IEEE Robotics and Automation Letters.

[25] W. Jallet, A. Bambade, S. El Kazdadi, C. Justin, and M. Nicolas, *Aligator*.

[26] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," en, *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, Dec. 2020.