

Torque-Minimizing Control Allocation for Overactuated Quadrupedal Locomotion

Mads Erlend Bøe Lysø* Esten Ingar Grøtli**
Kristin Ytterstad Pettersen*

* *Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway (e-mail: {mads.e.b.lyso,kristin.y.pettersen}@ntnu.no)*

** *Department of Mathematics and Cybernetics, SINTEF Digital, Trondheim, Norway (e-mail: Esten.Ingar.Grotli@sintef.no)*

Abstract: In this paper, we improve upon a method for optimal control of quadrupedal robots which utilizes a full-order model of the system. The original method utilizes offline nonlinear optimal control to synthesize a control scheme which exponentially orbitally stabilizes the closed-loop system. However, it is not able to handle the overactuated phases which frequently occur during quadrupedal locomotion as a result of the multi-contact nature of the system. We propose a modified method, which handles overactuated gait phases in a way that utilizes the full range of available actuators to minimize torque expenditure without requiring output trajectories to be modified. It is shown that the system under the proposed controller exhibits the same properties, i.e. exponential orbital stability, with the same or lower point-wise torque magnitude. A simulation study demonstrates that the reduction in torque may in certain cases be substantial.

Keywords: Robotics, Non-Linear Control Systems, Control Design, Optimal Control, Discrete Event and Hybrid Systems

1. INTRODUCTION

Quadrupedal robots have features that are beneficial for achieving dynamically stable and robust walking: Legged robots in general are able to adapt well to rough terrain while keeping the main base levelled, and the multi-contact nature of quadrupeds lend them some inherent robustness not shared by bipedal robots. However, the multi-contact interaction with its environment, combined with the high dimensionality of such systems, makes the modeling and control of such robots challenging. This has led to a greater reliance on clever heuristics and lower-order models when developing methods for control. As a result, progress has been made predominantly in simulations and in real-world experiments without a corresponding mathematical analysis or proof of theoretical stability. However, in recent years we have seen both increases in computational power and a greater understanding of and ability to exploit the structure of the control problem. As a result, it is increasingly feasible to utilize the full kinematic, and in some cases dynamical, models of the robot in offline and online methods for control. However, the constrained nature of these systems often leads to phases of the gait in which the ground contacts render the system dynamics overactuated. This complicates the control problem by making the series of inputs that correspond to a desired system behavior non-unique.

* This result is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, through the ERC Advanced Grant 101017697-CRÈME. The work is also supported by the Research Council of Norway through the Centres of Excellence funding scheme, project No. 223254 – NTNU AMOS.

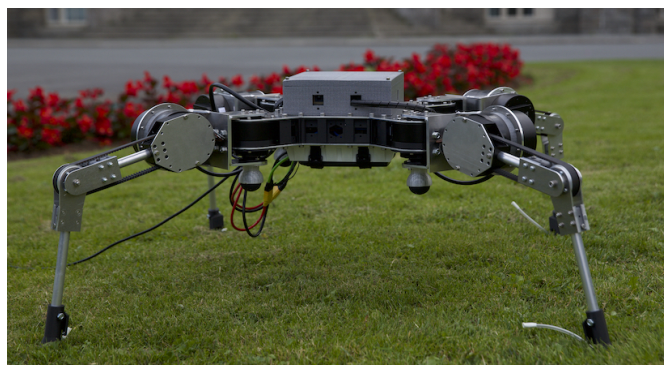


Fig. 1. Photo of the sprawling quadrupedal robot ASTRO. Several approaches have been taken in recent years to incorporate more detailed knowledge of the system dynamics into the control of quadrupedal robots. One popular avenue through which this has been explored is the control paradigm of model predictive control (MPC). While certain methods such as Di Carlo et al. (2018) have opted for fairly reduced-order models in order to yield linear MPC control schemes, others have attempted to more fully capture the complexities of the system dynamics. Neunert et al. (2018) utilize a full-order nonlinear model of the robotic system to produce a nonlinear model predictive control (NMPC) scheme. One of the advantages of this method is that the choice of gait pattern is also subject to online optimization, which is achieved through using a smooth approximation of the ground contact forces. In Minniti et al. (2022), a system model using the full kinematics and the dynamics of the base is utilized in an adap-

tive NMPC scheme. The scheme produces a state-feedback controller that asymptotically stabilizes the system to a desired trajectory in the presence of modeling errors. A quadratic programming (QP)-based whole body controller (WBC) is used to calculate the joint torques to track the desired accelerations and contact forces.

Another approach is to incorporate a high-fidelity robot model into an offline optimization problem. An example of this approach is presented in Ma et al. (2019). Here, the optimization problem describes a closed-loop system subject to a parametrized controller. The result of the optimization is a set of parameters define a closed-loop controller. Thus, no auxiliary control frameworks such as a WBC are needed to track the returned optimal trajectory. One benefit is that in contrast to e.g. a WBC-scheme, which may often require solving an optimization problem, the resulting controller here often has a much simpler structure. Specifically, the controller incorporated in Ma et al. (2019) is an input-output linearization controller, which exhibits properties such as Lipschitz continuity and continuous differentiability that may be important for stability analysis. The closed-loop system behavior is shown to be exponentially orbitally stable (EOS).

Finally, there have been some efforts in the later years to capture complex dynamics and scenarios without resorting to high-fidelity explicit modeling. Notably, in Miki et al. (2022), a method is presented utilizing deep reinforcement learning to produce an end-to-end model-free locomotion controller. The controller learns the most efficient trade-off between exteroception and proprioception for a wide range of scenarios. The empirical results of the paper show the robot traversing a mountain hiking route to the top faster than the nominal tourist traversal time.

While several of the above-mentioned methods yield good empirical results, most of them do not have theoretical guarantees for performance in the form of stability proofs. Even in Minniti et al. (2022), where a proof of convergence to a nominal trajectory is shown, this holds only for the simplified model. Also, stable trajectories for the span of the MPC horizon do not guarantee the generation of feasible, stable trajectories in the long run. In this respect, the controller presented in Ma et al. (2019) seems to have a clear advantage: The synthesized controller stabilizes the full dynamics exponentially to a submanifold of the state space, which again contains an EOS orbit.

However, the method as described does not cope satisfactorily with the following challenge: During some phases of a gait cycle, the robot has two legs in contact with the ground. During these phases the robot is over-actuated, and there is an infinite set of inputs that will result in exactly the same acceleration. Ma et al. (2019) handles this problem by disabling one actuator in such cases, returning the system to a state of non-overactuation. This is not necessarily a satisfying solution. Firstly, the choice of which actuator to disable is not explicitly justified, and it is not clear what method one would use to make this decision. Secondly, the analogy would be for a human standing on two legs to disable its thigh muscles to avoid overactuation, which is not obviously the most robust nor energy-efficient answer to the question of torque-allocation. One way to handle this issue could be to simply opt for e.g.

a QP-based WBC, which is capable of choosing from this set of possible inputs subject to some objective, e.g. minimization of contact forces or torques. However, the simple, transparent structure of a closed-form controller would then be lost. Moreover, properties such as Lipschitz continuity are not immediately present in such kinds of controllers, which may interfere with the stability analysis.

In this paper, we modify the method presented in Ma et al. (2019) to achieve point-wise optimal (in a least-squares sense) control allocation for a given desired output trajectory. Specifically, we alter the control structure to allow for a greater number of actuators than actuated degrees of freedom (DOFs). Thus, instead of removing the resulting ambiguity in the choice of inputs by disabling the superfluous actuators, our controller utilizes this ambiguity to choose the smallest of the inputs which realize the output trajectory. As in Ma et al. (2019), the resulting closed-loop system behavior exhibits exponential stability of the outputs, and exponential orbital stability of the full system state modulo horizontal position. Furthermore, the resultant torque expenditure is point-wise less than or equal to the torque expenditure of the previous method, without requiring any modification of the system trajectories. A simulation study is presented, and this demonstrates that the torque reduction is fairly substantial in certain cases, as is the reduction in energy expenditure that follows.

The paper is organized as follows. In Section 2 we briefly describe the dynamic system formulation. In Section 3 we first describe the optimal control formulation and the time-varying input-output linearization scheme as is used in Ma et al. (2019), as well as the method for exponential stabilization, which we build upon. Then, we present the modified controller which readily handles overactuated phases of the gait. We then present simulation results illustrating the proposed method and showing the stability and energy-expenditure in Section 4. Finally, in Section 5 we present the conclusions and discuss future work.

2. SYSTEM MODELING

In this section, we briefly describe the model of the system. The robotic system to be modeled is the quadrupedal robot ASTRO, short for Articulated Sprawling Tetrapod Robot, see Fig. 1. The robot has a floating base which is not directly actuated, as well as four legs, each with three actuated joints. Each leg first has a hip link connected to the base by a joint whose rotational axis is normal to the dorsal plane. Then, the upper and lower leg links are connected in series with joints whose rotational axes lie in the dorsal plane of the robot. The base has 3 positional and 3 orientational DOFs, with the orientation parametrized by Tait-Bryan angles, while each leg has 3 joints. In total, this results in 18 DOFs for the system. The generalized coordinates become $\mathbf{q} = \left(\mathbf{p}_b^\top, \phi_b^\top, \boldsymbol{\theta}_{\text{legs}}^\top \right)^\top$, where \mathbf{p}_b and ϕ_b represent the position and orientation of the base, while $\boldsymbol{\theta}_{\text{legs}}$ represents the joint angles of the legs. The following transcription of the ASTRO robot into a hybrid dynamical system (HDS) is based on work found in Hereid et al. (2018); Ma et al. (2019), but there performed respectively for a bipedal and mammalian quadrupedal robot.

2.1 Hybrid Dynamical System Formulation

A walking robot can be described as a HDS. Adhering to the formulation in Hamed et al. (2019), the HDS can be

described by a set of continuous system dynamics $FG = \{(f_v, g_v)\}$ on a set of continuous domains $\mathcal{D} = \{\mathcal{D}_v\}$ so that $\dot{\mathbf{x}}_v = f_v(\mathbf{x}_v) + g_v(\mathbf{x}_v)\mathbf{u}_v$ where \mathbf{x}_v is the state and \mathbf{u}_v is the input, and $(\mathbf{x}_v, \mathbf{u}_v) \in \mathcal{D}_v$. Each domain and system of equations correspond to a vertex $v \in \mathcal{V}$ of a graph $\Lambda = (\mathcal{V}, \mathcal{E})$, with \mathcal{E} being the set of edges. In our case we denote the state $\mathbf{x}_v = (\mathbf{q}_v^\top, \dot{\mathbf{q}}_v^\top)^\top$. We define the successor function $\mu : \mathcal{V} \rightarrow \mathcal{V}$ so that $v_2 = \mu(v_1)$ if there is an edge $(v_1 \rightarrow v_2) \in \mathcal{E}$. Transitions between domains are governed by a set of guards $\mathcal{S} = \{\mathcal{S}_{(v \rightarrow \mu(v))}\}_{(v \rightarrow \mu(v)) \in \mathcal{E}}$ and discrete dynamics $\Delta = \{\Delta_{(v \rightarrow \mu(v))}\}_{(v \rightarrow \mu(v)) \in \mathcal{E}}$. Each guard is a surface in \mathcal{D}_v so that a transition occurs from v to $\mu(v)$ when $\mathbf{x}_v \in \mathcal{S}_{(v \rightarrow \mu(v))}$. The transition occurs according to $\mathbf{x}_{\mu(v)} = \Delta_{(v \rightarrow \mu(v))}(\mathbf{x}_v)$.

2.2 Continuous dynamics and constraints

We introduce the index set $\mathcal{I}_{\text{legs}} = \{\text{fl}, \text{rl}, \text{fr}, \text{rr}\}$ and refer to the vector of joint angles for leg i as $\boldsymbol{\theta}_i$, e.g. $\boldsymbol{\theta}_{\text{fl}}$ for the front left leg. Furthermore, the robot has some of its legs in contact with the ground at any given time, encoded by the index set $\mathcal{I}_c \subseteq \mathcal{I}_{\text{legs}}$. We here model the feet as point feet, and assume a positional no-slip condition and zero rotational friction. The first is a standard assumption in the literature, the second is a modeling choice. In Hereid et al. (2018) a rotational no-slip condition on the surface normal axis is enforced as well. However, the frequent multi-contact phases and the sprawling configuration is likely to make such a condition too restrictive for our robot due to the alignment between the hip joint rotational axis and the surface normal, which is not present in the mammalian configuration. The no-slip conditions are described by a vector of kinematic constraints

$$\mathbf{g}_c(\mathbf{q}) = \text{vec}(\{\mathbf{p}_i(\mathbf{q}) - \mathbf{p}_i(\mathbf{q}(0)) = \mathbf{0}\}_{i \in \mathcal{I}_c}) \quad (1)$$

with $\mathbf{p}_i(\mathbf{q})$ denoting the position of the point-foot i in the world frame. From the Euler-Lagrange equation we get our equations for the constrained nonlinear dynamical system:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{B}\mathbf{u} + \mathbf{J}_c^\top(\mathbf{q})\boldsymbol{\lambda} \quad (2a)$$

$$\mathbf{J}_c(\mathbf{q})\ddot{\mathbf{q}} + \frac{\partial}{\partial \mathbf{q}}(\mathbf{J}_c(\mathbf{q})\dot{\mathbf{q}}) = \mathbf{0} \quad (2b)$$

where $\mathbf{B} = [\mathbf{0}_{6 \times 12}^\top, \mathbf{I}_{12}]^\top$ is the actuation matrix, $\mathbf{D}(\mathbf{q})$ is the mass matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is the Coriolis matrix, and $\mathbf{G}(\mathbf{q})$ represents the potential terms. The Lagrange multipliers of the constraint forces are denoted by $\boldsymbol{\lambda}$, while $\mathbf{J}_c(\mathbf{q}) = \frac{\partial}{\partial \mathbf{q}}\mathbf{g}_c(\mathbf{q})$. We may re-arrange these equations to the explicit, control-affine form assumed in the HDS formulation:

$$\dot{\mathbf{x}}_v = \mathbf{f}_v(\mathbf{x}_v) + \mathbf{g}_v(\mathbf{x}_v)\mathbf{u}_v \quad (3)$$

2.3 Discrete dynamics and transitions

Each discrete transition between subsystems represents either the landing or lift-off of a foot. For transitions that correspond to lift-offs, the discrete dynamics are simply the identity. For transitions corresponding to feet hitting the ground, however, we assume a perfectly plastic impact; this is a typical choice in the literature to avoid the identification of parameters related to elastic impact (Grizzle et al., 2014, p. 12). This implies that there is no abrupt change in \mathbf{q} , but the velocity of the impact foot immediately becomes 0. From this and the conservation of momentum, we get the discrete dynamics

$$\Delta_{(v \rightarrow \mu(v))}(\mathbf{x}_v) = \begin{pmatrix} \dot{\mathbf{q}}_v + \mathbf{D}^{-1}(\mathbf{q}_v)\mathbf{J}_{\mu(v)}(\mathbf{q}_v)\boldsymbol{\lambda}_{\text{impulse}} \\ \mathbf{q}_v \end{pmatrix} \quad (4)$$

for impact, where $\boldsymbol{\lambda}_{\text{impulse}}$ is the intensity of the contact impulse required to set all foot velocities to 0. For lift-off, the dynamics simply become $\Delta_{(v \rightarrow \mu(v))}(\mathbf{x}_v) = \mathbf{x}_v$.

2.4 Guards and admissible domain

For each vertex, the admissible domain \mathcal{D} is described as the states satisfying a set of constraints. Firstly, there are constraints relating to the contact forces, denoted $\mathbf{v}_v(\mathbf{q}_v, \dot{\mathbf{q}}_v)\boldsymbol{\lambda}_v(\mathbf{q}_v) \geq \mathbf{0}$. Specifically, the contact force must not exit the linearized friction cone so as to not slip. Secondly, there are constraints independent of the contact forces. We denote these as $\mathbf{h}_v(\mathbf{q}_v, \dot{\mathbf{q}}_v) \geq \mathbf{0}$. Here, we require the position of all swing feet to be greater than 0, i.e., above the ground. These constraints are summarized as

$$\mathbf{A}_v = \begin{bmatrix} \mathbf{v}_v(\mathbf{q}_v, \dot{\mathbf{q}}_v)\boldsymbol{\lambda}_v(\mathbf{q}_v) \\ \mathbf{h}_v(\mathbf{q}_v, \dot{\mathbf{q}}_v) \end{bmatrix} \geq \mathbf{0} \quad (5)$$

Each guard is the subset of the boundary of the domain, and a transition occurs when the state is about to exit the domain through it. Thus, for a select element $H_{(v \rightarrow \mu(v))}$ from (5), we may define a corresponding guard as

$$\mathcal{S}_{(v \rightarrow \mu(v))} = \left\{ (\mathbf{q}_v, \dot{\mathbf{q}}_v, \mathbf{u}_v) \left| \begin{array}{l} H_{(v \rightarrow \mu(v))}(\mathbf{q}_v, \dot{\mathbf{q}}_v) = 0 \\ \dot{H}_{(v \rightarrow \mu(v))}(\mathbf{q}_v^-, \dot{\mathbf{q}}_v^-) < 0 \end{array} \right. \right\} \quad (6)$$

with $\dot{H}_{(v \rightarrow \mu(v))}(\mathbf{q}_v^-, \dot{\mathbf{q}}_v^-)$ denoting the left hand derivative with respect to time. For transitions corresponding to a foot lift-off, $H_{(v \rightarrow \mu(v))}$ is the normal component of the contact force of the lift-off foot. For transitions corresponding to an impact, $H_{(v \rightarrow \mu(v))}$ is the impact foot height.

3. METHODS

In this section we briefly describe the methods we build on from Ma et al. (2019), before we introduce our suggested modification to handle overactuated phases.

3.1 Offline optimal control

As in Ma et al. (2019) we use the Fast Robotics Optimization and Simulation Toolkit (FROST), first presented in Hereid and Ames (2017), to perform offline optimal control of the HDS described in Section 2. For the sake of the optimization problem each domain has only one possible successor, and the domains in this cycle determines the gait pattern. The duration of each phase is subject to optimization. Periodicity is enforced by the boundary condition

$$\mathbf{x}(t_0) + (t_f - t_0)\mathbf{v}_{\text{des,xy}} - \Delta_{v_f \rightarrow \mu(v_f)}(\mathbf{x}(t_f)) = \mathbf{0} \quad (7)$$

where v_f is the final vertex of Λ , and $\mathbf{v}_{\text{des,xy}}$ is nonzero in the first two elements and describe the desired horizontal velocity. t_0 and t_f denote the initial and final times.

FROST utilizes a direct collocation transcription scheme and pre-compiled symbolic representations of constraints to make optimization over high-dimensional and highly complex systems computationally feasible. The framework is designed to incorporate control laws into the optimization problem itself, so that solution contains a set of controller parameters $\boldsymbol{\alpha}^*$ which define a control law $\mathbf{u}(t, \mathbf{x}, \boldsymbol{\alpha}^*)$. Please see Hereid et al. (2018) for a detailed description of FROST. The objective function is in this case chosen as the sum of the squared 2-norm of torque expenditure, divided by the period of the resulting gait. The division is done so as to avoid the solver favoring shorter gaits.

The discretized optimization problem then becomes

$$\min_{\{\mathbf{x}_k\}, \{\mathbf{u}_k\}, \{\boldsymbol{\alpha}_k\}, \{t_k\}} \frac{1}{t_f - t_0} \sum_k \mathbf{u}_k^\top \mathbf{u}_k$$

s.t.

System dynamics as per (2) and (4)(transcribed) (8)

Chosen control law $\mathbf{u}(t, \mathbf{x}, \boldsymbol{\alpha})$

State constraints including (5)

Boundary conditions (7)

where $k \in 0 \dots N - 1$ is the discretization node index, N being the number of discretization nodes.

3.2 IO-linearization and zero-dynamics

As in the case of Ma et al. (2019), the controller we incorporate is an input-output linearization controller. Consider the system (3) with n states and m inputs. We now add an output $\mathbf{y}_a(\mathbf{x}) = \mathbf{h}(\mathbf{x})$ of dimension n_y , which we denote the actual output. We then define $\mathbf{y}_d(t, \boldsymbol{\alpha})$, a time-varying vector of desired outputs. Following Ma et al. (2019), these are described as 4th-order Bézier curves, parametrized by $\boldsymbol{\alpha}$. Lastly, we define $\mathbf{y}(t, \mathbf{x}, \boldsymbol{\alpha}) = \mathbf{y}_a(\mathbf{x}) - \mathbf{y}_d(t, \boldsymbol{\alpha})$ so that $\mathbf{y}(t, \mathbf{x}, \boldsymbol{\alpha}) = 0$ whenever $\mathbf{y}_a(\mathbf{x}) = \mathbf{y}_d(t, \boldsymbol{\alpha})$. The behavior of $\mathbf{y}(t, \mathbf{x}, \boldsymbol{\alpha})$ is referred to as the output-dynamics, while the behavior of the system when $\mathbf{y}(t, \mathbf{x}, \boldsymbol{\alpha}) \equiv 0$ is denoted the zero-dynamics.

As described in Henson and Seborg (1997), if the system has a well-defined *vector relative degree* $\{r_i\}_{i \in 1 \dots n_y}$, it may be transformed into an equivalent system where a subset of the states are the outputs of the original system along with their derivatives up to r_i . We may now define $\mathbf{y}_r \triangleq (y_1^{(r_1-1)}, \dots, y_{n_y}^{(r_{n_y}-1)})$. In order to achieve the linear output-dynamics $\dot{\mathbf{y}}_r = \mathbf{v}$, given the assumption of well-defined relative degree, we choose the input \mathbf{u} to be

$$\mathbf{u}(t, \mathbf{x}, \boldsymbol{\alpha}) = \mathcal{A}^{-1}(\mathbf{x}) (-\mathbf{b}(\mathbf{x}) + \mathbf{y}_{d,r}(t, \boldsymbol{\alpha}) + \mathbf{v}) \quad (9a)$$

where

$$\mathbf{b}(\mathbf{x}) = \left(L_{\mathbf{f}}^{r_1} h_1(\mathbf{x}), \dots, L_{\mathbf{f}}^{r_{n_y}} h_{n_y}(\mathbf{x}) \right)^\top \quad (9b)$$

$$\mathbf{y}_{d,r}(t, \boldsymbol{\alpha}) = \left(y_{d,1}^{(r_1)}(t, \boldsymbol{\alpha}), \dots, y_{d,i}^{(r_i)}(t, \boldsymbol{\alpha}) \right)^\top \quad (9c)$$

$$\mathcal{A}(\mathbf{x}) = \begin{bmatrix} L_{\mathbf{g}_1} L_{\mathbf{f}}^{r_1-1} h_1(\mathbf{x}) & \dots & L_{\mathbf{g}_m} L_{\mathbf{f}}^{r_1-1} h_1(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ L_{\mathbf{g}_1} L_{\mathbf{f}}^{r_{n_y}-1} h_{n_y}(\mathbf{x}) & \dots & L_{\mathbf{g}_m} L_{\mathbf{f}}^{r_{n_y}-1} h_{n_y}(\mathbf{x}) \end{bmatrix} \quad (9d)$$

We refer to $\mathcal{A}(\mathbf{x})$ as the decoupling matrix. The full sets of equations can be found in e.g. Henson and Seborg (1997, pp. 160-164), although we here express the output-dynamics in original coordinates in the interest of brevity.

We opt for an ambling gait where two legs are in contact with the ground at any point, and as such the system has 11 actuated DOFs (all leg joints with the constraint of constant distance between stance legs). The method requires an equal number of outputs and actuated DOFs (Hereid et al. (2018)). In the interest of a most direct comparison with the method of Ma et al. (2019), we choose the leg joints except the rear stance hip pitch as outputs.

3.3 Exponential Orbital Stabilization of full state

Although the IO-linearization controller stabilizes the output-dynamics exponentially, the system as a whole may still be unstable. It is desirable that the closed-loop system has an exponentially stable orbit in the state modulo

the horizontal position, as the robot should move forward. The stability analysis of an orbit can be performed by considering the state trajectory at intersections with a transverse surface as a discrete dynamical system, described by a Poincaré map. The orbit will be EOS when the spectral radius of the linearization of the map around the point of intersection is less than 1 (Hamed et al., 2015).

As in Ma et al. (2019), we now consider the outputs \mathbf{y} in the optimized controller as linear combinations of system states and corresponding optimal trajectories, parametrized by weights $\boldsymbol{\theta}$, i.e. $\mathbf{y}(t, \mathbf{x}, \boldsymbol{\alpha}^*, \boldsymbol{\theta})$. The outputs chosen during gait generation gives our initial guess $\boldsymbol{\theta}_0$. We then formulate the problem of exponential orbital stabilization as a bilinear matrix inequality-constrained optimization problem in $\boldsymbol{\theta}$. Solving this optimization problem leaves us with a set of weights $\boldsymbol{\theta}^*$ determining $\mathbf{y}(t, \mathbf{x}, \boldsymbol{\alpha}^*, \boldsymbol{\theta}^*)$ which makes the resulting gait EOS (modulo horizontal position). See Hamed et al. (2019, 2015) for further details and proof of the resulting stability.

3.4 Over-actuated input-output linearization

In order for $\mathcal{A}(\mathbf{x})$ in (9a) to be invertible we must have $m = n_y$. Now, consider the case where $m > n_y$, i.e. the robot is overactuated. ASTRo, which is a quadrupedal robot, will typically have two stance feet during some phases of a gait. During these phases the number of outputs, which must equal the number of actuated DOFs, is 11, while $m = 12$. As a result, the output-dynamics of the IO-linearized system are over-actuated. In Ma et al. (2019) this discrepancy is resolved by disabling one of the inputs, so that $m = 11$. We here suggest a different approach.

We will assume that the now non-square $\mathcal{A}(\mathbf{x}) \in \mathbb{R}^{n_y \times m}$ has constant rank n_y , which is just to say that at any point in time there is a subset of inputs which can be removed to recoup the system properties assumed in the case where $n_y = m$. We can choose the feedback-law as

$$\mathbf{u}(t, \mathbf{x}) = \mathcal{A}^+(\mathbf{x}) (-\mathbf{b}(\mathbf{x}) + \mathbf{y}_{d,r}(t) + \mathbf{v}) \quad (10)$$

where $(\cdot)^+$ denotes the Moore-Penrose pseudoinverse (MPP), to again achieve $\dot{\mathbf{y}}_r = \mathbf{v}$ as desired.

Most importantly, (10) yields exactly the linear, decoupled output-dynamics in the overactuated case, as (9a) does in the non-overactuated case. This is due to the system of linear equations which \mathcal{A}^+ solves being underdetermined rather than overdetermined, and thus admitting of infinitely many exact solutions.

Secondly, recall that the matrix $\mathcal{A}(\mathbf{x})$ is assumed to have the constant rank n_y . As a result, the MPP is a continuous and continuously differentiable function of the original matrix (Golub and Pereyra, 1973). Thus, assuming continuity and continuous differentiability of $\mathcal{A}(\mathbf{x})$ with respect to \mathbf{x} , these properties also hold for $\mathcal{A}^+(\mathbf{x})$. As a result, the control law $\mathbf{u}(t, \mathbf{x})$ is continuously differentiable in \mathbf{x} for an appropriate choice of \mathbf{v} , a typical choice being a linear combination of elements from \mathbf{y} and their derivatives up to respective relative degree.

Finally, as the MPP characterizes the least-squares solution to an under-determined linear system, the modified controller gives the point-wise smallest control signal that results in the desired output-dynamics in the least-squares sense. In the case where the input signal is a torque, this

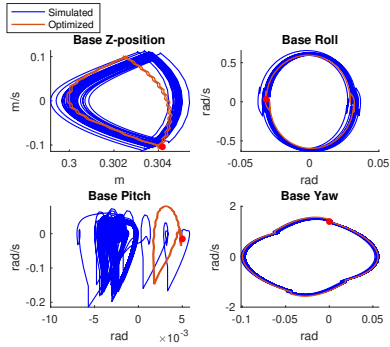


Fig. 2. Phase portraits for base of optimized and simulated ambulating gait. Red dots signify initial values.

results in a controller which utilizes all available actuators to minimize torque expenditure along the trajectory.

The minimization in the least-squares sense is also particularly desirable for the following reason. The energy expenditure of electric actuators can be divided into mechanical energy and Joule heating, which was in the case of Seok et al. (2015) found to account for 75% of expended energy, is well approximated as proportional to the square of applied torque. Thus, it is expected that minimizing the applied torque in a least-squares sense will reduce the expended energy as a result.

It should be noted here that the identical closed-loop system behavior in general holds only with respect to the output-dynamics; the effects on the unobservable states are not discussed. However, the design of the controller in the conventional case is performed only so as to exponentially stabilize the output-dynamics, and the stability of the zero-dynamics is subject to further analysis in any case. Just as in the case of Ma et al. (2019), the post-processing described in Section 3.3 ensures that the dynamics are EOS along the zero-dynamics manifold.

4. RESULTS

We now present the results with regards to energy and torque expenditure, where we compare between the method as described in Ma et al. (2019) and the method with our modified controller as described in Section 3.4.

We solve the optimization problem posed in (8), using 0.2 m s^{-1} as the forward velocity in the boundary condition, and choosing the HDS graph to produce an ambulating gait seen in the diagram in Fig. 4. The controller in (10) is used during optimization. A post-processing of the outputs is performed as described in Section 3.3. During this procedure the spectral radius of the linearized Poincare map of the orbit goes from 1.2024 to 0.6267, rendering the orbit EOS. The resulting set of parameters α^* and θ^* define our outputs \mathbf{y} for the controller. The gait is then simulated for 500 gait cycles using FROST. The phase portraits for the optimized and simulated state trajectories are shown in Figs. 2 and 3. Trajectories for the x - and y -positions do (by design) not constitute orbits and are omitted.

The system is simulated using both the method proposed by Ma et al. (2019), where 11 actuators are utilized, and with the controller (10) proposed in Section 3.4, allowing 12 actuators to be utilized. In Table 1 the peak and root mean square (RMS) torques, as well as the cost of transport (CoT) are reported. The CoT is defined as

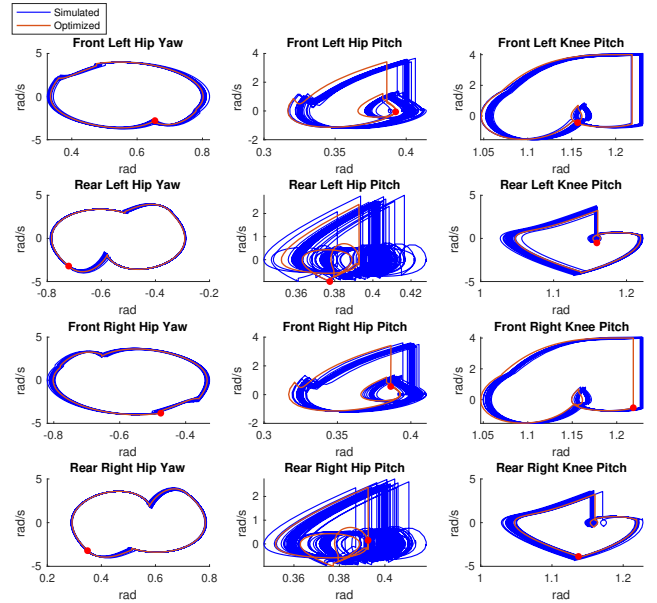


Fig. 3. Phase portraits for joints of optimized and simulated ambulating gait. Red dots signify initial values.

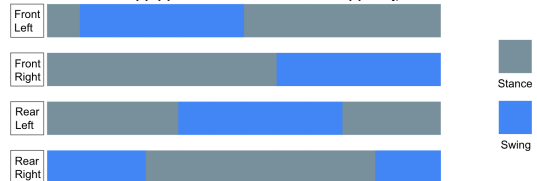


Fig. 4. Diagram for one cycle of an ambulating gait.

$\frac{E}{mgd}$ where E is the expended energy, m and g are the mass of the robot and the gravitational constant and d is the total distance traveled. The following choices for modeling energy expenditure are made: As in Seok et al. (2015), we calculate the power expenditure of each actuator as the sum of expended mechanical power and power lost to heating effects. For the mechanical energy, we choose a simple model with no recuperation of absorbed mechanical energy through e.g. regenerative braking. We thus calculate expended mechanical power as $P_{\text{mec}} = \max(\tau\omega, 0)$ for each motor, where τ is the joint torque and ω is the angular velocity. The power lost to heating effects as a function of the joint torque is written as $P_{\text{heat}} = R \left(\frac{\tau_{\text{motor}}}{K_t} \right)^2$, where K_t is the motor torque constant, R is the resistance of the motor and τ_{motor} is the motor torque. The motor torque is defined as $\tau_{\text{motor}} = \frac{\tau}{N}$ where N is the gear ratio between the motor and the joint. For calculating heating effects as a function of joint torques, we require properties of a specific actuator which is not a part of the model used in simulation. In the interest of performing these calculations with realistic parameter values, we base ourselves on the RMD-X8-Pro-H from MyActuator, with an additional gear ratio of 6:1, i.e. $N = 6$. The actuator has a $K_t = 0.29 \text{ N m A}^{-1}$ and an $R = 0.55 \Omega$. The gear ratio is chosen to bring the torque trajectories within the nominal torque operating range of the actuator while retaining the resulting angular velocities within the geared angular velocity operating range. The CoT can then be written

$$\text{CoT} = \frac{1}{mgd} \int_{t_0}^{t_f} \sum_{\text{actuators}} (P_{\text{mec}}(t) + P_{\text{heat}}(t)) dt \quad (11)$$

# Method	CoT	Peak torque	Torque RMS
Ma et al. (2019)	6.389	83.97 N m	9.659 N m
Proposed	4.532	66.55 N m	7.932 N m

Table 1. Performance metrics for previous and proposed method

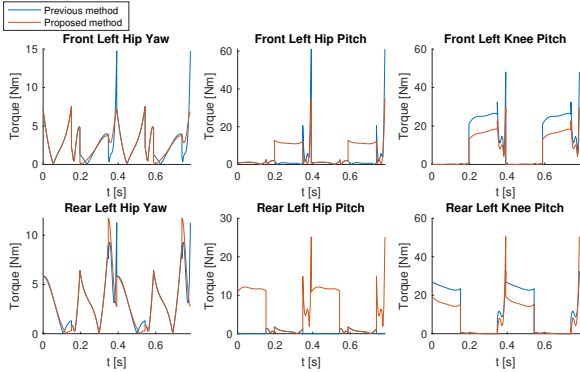


Fig. 5. Plots of absolute value of torques for left leg actuators for two consecutive gait cycles.

As we can see in Table 1, our proposed method yields a fairly substantial reduction in both the CoT, peak torque and RMS torque. The CoT is reduced by $\approx 29.1\%$. The RMS torque is reduced by $\approx 17.9\%$, and the peak torque is reduced by $\approx 20.7\%$. This indicates that the modification, in addition to avoiding arbitrary choices of which inputs to disable, may actually lead to fairly large energy savings in practice, without at all having to modify the output trajectories of the system.

In Fig. 5, in order to investigate the differences in the torque profiles more closely, we have plotted the absolute value of the torques of the left leg actuators for both methods. Due to the symmetry of the ambling gait, the right leg actuator torques will be identical but phase-shifted by half a gait cycle. As can be seen, the rear hip pitch torque is much greater for our method compared to the method proposed in Ma et al. (2019). This is to be expected: In the original method the rear hip pitch actuators are disabled during stance phases to avoid overactuation. Thus, the rear hip pitch actuators are only active during swing leg phases, in which the actuator only lifts the leg off the ground. The front hip pitch actuator is also seen to be more active for the proposed method, although the picture is not as clear since the method from Ma et al. (2019) exhibits significantly higher peak torques. The increased torque use in hip pitch actuators in our method, however, is seen to lead to substantially lower torque expenditure in all knee actuators compared to the method from Ma et al. (2019). This even distribution of load-bearing between actuators, along with heat-related power losses depending on the square of torque expenditure, explains the significant reduction in CoT.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a modification to an existing method for controller synthesis through offline nonlinear optimal control. The modification allows the controller to better utilize all available inputs in overactuated scenarios. Simulations indicate that the savings in torque expenditure and CoT may be quite significant in certain cases when compared to the original method, without requiring any modification of the system trajectories.

Future work includes experimental validation of the torque reduction which we have here observed in simulations, as well as exploring further modifications to the controller with the aim of increasing the robustness.

REFERENCES

- Di Carlo, J., Wensing, P.M., Katz, B., Bledt, G., and Kim, S. (2018). Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control. In *Proc. 2018 IEEE/RSJ Int. Conf. Intelligent Robots and Systems, Madrid, Spain*.
- Golub, G.H. and Pereyra, V. (1973). The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM J. on Numerical Analysis*, 10(2), 413–432.
- Grizzle, J.W., Chevallereau, C., Sinnet, R.W., and Ames, A.D. (2014). Models, feedback control, and open problems of 3D bipedal robotic walking. *Automatica*, 50(8), 1955–1988.
- Hamed, K.A., Buss, B., and Grizzle, J. (2015). Exponentially Stabilizing Continuous Time Controllers for Periodic Orbits of Hybrid Systems: Application to Bipedal Locomotion with Ground Height Variations. *Int. J. of Robotics Research*, accepted to appear.
- Hamed, K.A., Wen-Loong, and Ames, A.D. (2019). Dynamically Stable 3D Quadrupedal Walking with Multi-Domain Hybrid System Models and Virtual Constraint Controllers. In *Proc. 2019 American Control Conference, Philadelphia, PA, USA*.
- Henson, M.A. and Seborg, D.E. (1997). *Nonlinear Process Control*. Prentice Hall PTR.
- Hereid, A. and Ames, A.D. (2017). FROST: Fast robot optimization and simulation toolkit. In *Proc. 2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Vancouver, Canada*.
- Hereid, A., Hubicki, C.M., Cousineau, E.A., and Ames, A.D. (2018). Dynamic Humanoid Locomotion: A Scalable Formulation for HZD Gait Optimization. *IEEE Trans. on Robotics*, 34(2), 370–387.
- Ma, W.L., Hamed, K.A., and Ames, A.D. (2019). First Steps Towards Full Model Based Motion Planning and Control of Quadrupeds: A Hybrid Zero Dynamics Approach. In *Proc. 2019 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Macau, China*.
- Miki, T., Lee, J., Hwangbo, J., Wellhausen, L., Koltun, V., and Hutter, M. (2022). Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62).
- Minniti, M.V., Grandia, R., Farshidian, F., and Hutter, M. (2022). Adaptive CLF-MPC With Application to Quadrupedal Robots. *IEEE Robotics and Automation Letters*, 7(1), 565–572.
- Neunert, M., Stäuble, M., Gifftthaler, M., Bellicoso, C.D., Carius, J., Gehring, C., Hutter, M., and Buchli, J. (2018). Whole-Body Nonlinear Model Predictive Control Through Contacts for Quadrupeds. *IEEE Robotics and Automation Letters*, 3(3), 1458–1465.
- Seok, S., Wang, A., Chuah, M.Y., Hyun, D.J., Lee, J., Otten, D.M., Lang, J.H., and Kim, S. (2015). Design Principles for Energy-Efficient Legged Locomotion and Implementation on the MIT Cheetah Robot. *IEEE/ASME Trans. on Mechatronics*, 20(3), 1117–1129.