

A Computational Framework for Designing Skilled Legged-wheeled Robots

Moritz Geilinger¹, Sebastian Winberg¹, Stelian Coros¹

Abstract—Legged-wheeled robots promise versatile, fast and efficient mobile capabilities. To unleash their full potential, however, such hybrid robots need to be designed in a way that promotes the complex, full-body motions required for novel locomotion modes. This paper discusses the computational framework we have used to create a new type of legged robot which, when equipped with different types of end effectors, is capable of an array of interesting locomotory behaviors, including walking, roll-walking, roller-blading, and ice-skating. We show that this computational framework, which builds on a design system we recently introduced in the computer graphics community [1], can accurately predict the way in which different design decisions affect the robot’s ability to move, thus serving as an important tool in engineering new types of mobile robots. We also propose a novel warm-starting method which leverages ideas from numerical continuation to drastically improve convergence rates for the trajectory optimization routine we employ to generate optimal motions.

Index Terms—Legged Robots, Wheeled Robots, Optimization and Optimal Control, Motion and Path Planning **TODO: List of keywords (from the RA Letters keyword list)**

I. INTRODUCTION

LEGGED-WHEELED robots hold the promise to deliver dexterous mobility while at the same time being fast and efficient. Combining legs and wheels, however, poses interesting new challenges: the morphology of a robot determines the space of motions the robot can perform, and different types of end effectors govern the way in which it can locomote. This relationship between mechanical design and motion capabilities becomes especially complex when we want to exploit the combined advantage of legs and wheels. Due to these benefits and challenges, legged-wheeled locomotion has become an increasingly active area of research in recent years. While most advances in legged-wheeled locomotion have focused on studying predefined robot designs and their locomotion modes in isolation, our long term goal is to develop a general computational framework that will serve as a powerful tool in engineering new types of hybrid mobile robots. To this end, we evaluate the predictive power of our

Manuscript received: September, 11, 2019; Revised December, 27, 2020; Accepted February, 8, 2020.

This paper was recommended for publication by Editor Nikos Tsagarakis upon evaluation of the Associate Editor and Reviewers’ comments.

¹Moritz Geilinger, Sebastian Winberg and Stelian Coros are with the Computational Robotics Lab, ETH Zurich, Switzerland.

moritzge@inf.ethz.ch, winbergs@student.ethz.ch, scoros@inf.ethz.ch

This work was supported in part by the Swiss National Science Foundation through the National Centre of Competence in Digital Fabrication (NCCR dfab).

Digital Object Identifier (DOI): see top of this page.



Fig. 1. AgileBot and SkaterBot, two robots created with the help of our interactive design system.

recent design and motion synthesis system [1] for legged-wheeled robots. On a technical level, we introduce a novel warm-starting technique to drastically improve the reliability and convergence of the trajectory optimization routine used to synthesize locomotion behaviors. This technique builds on numerical continuation ideas, and it is particularly effective for motions that combine stepping and rolling behaviors. We also build and analyze a novel legged-wheeled robot to validate our computational framework. Equipped with different types of end effectors (i.e. actuated or unactuated wheels), this robot is capable of a variety of interesting locomotion modes that exploit the combined advantages offered by limbs and wheels.

A. Related Work

Motion planning and control for legged robots has a rich history in the field of robotics, given the promise that such machines have when it comes to navigation in unstructured environments. Nevertheless, wheeled and tracked robotic systems enjoy widespread use in industry, as they are more efficient, more stable and easier to control reliably. It comes as no surprise, though, that researchers have also focused their efforts on the challenge of creating hybrid robots that combine the advantages of wheels and legs. For example, the robot presented in [2] uses a manually designed gait that exploits limb articulation and unactuated wheels to create a very particular

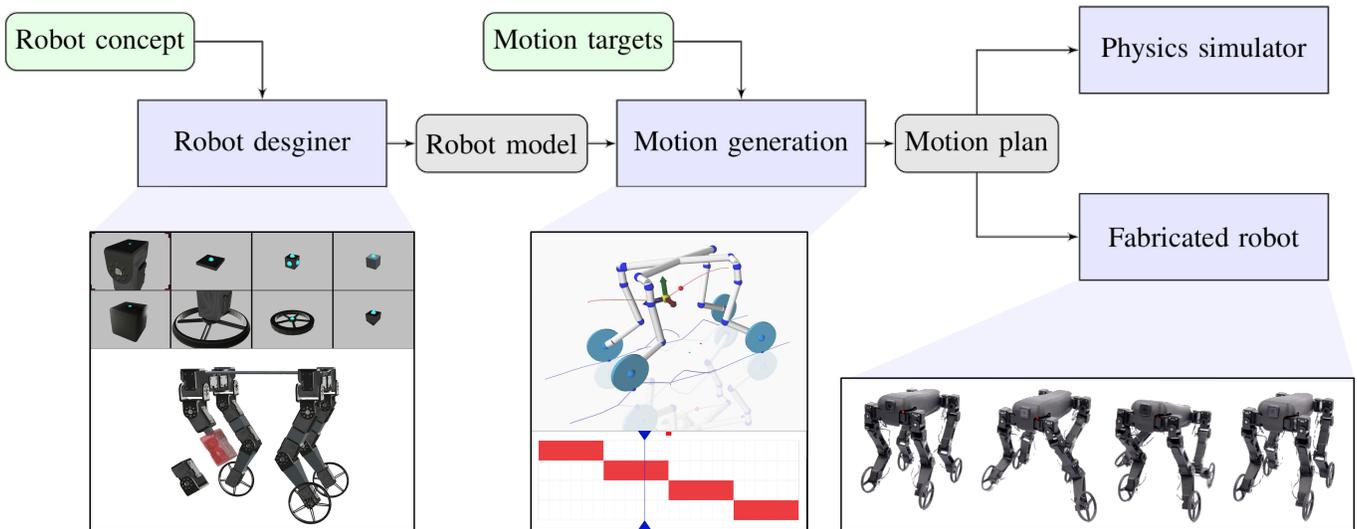


Fig. 2. This figure depicts the computational framework we use to design hybrid legged/wheeled robots: Left: Users can generate arbitrary robot designs by mixing together different building blocks using an interactive graphical interface. Middle: Physically-correct locomotion behaviors are generated through trajectory optimization. Right: These motions can then either be tested in simulation, or they can be transferred wirelessly to the physical robot.

type of wheeled locomotion. Another interesting robot design that combines legs and wheels is presented in [3]. Here, the robot uses its legs to actively control its support polygon while driving with motorized wheels, which enables it to perform aggressive start-stop and turning motions. Boston Dynamics’ Handle robot [4], one of the most recent examples of a hybrid legged/wheeled system, displays an incredibly rich repertoire of agile behaviors, although the technical details behind this platform and its control system remain unpublished.

Traditional legged robot designs are well-studied. However, equipping them with various types of end effectors opens up interesting avenues for further investigations. For example, the quadrupedal robot Anymal was recently retro-fitted with different types of wheels [5], [6], and even with ice skates [7]. The Anymal team also developed tailored techniques for motion planning and full body control for the resulting mechatronic systems. Their efforts in this area highlight an important challenge. While Anymal was able to move using motorized wheels reasonably well [5], when equipped with ice skates, it had very limited mobility [7]. We hypothesize that this important limitation is due to Anymal’s morphological design, which was intended for traditional legged locomotion: the types of end effector trajectories needed for ice-skating might be difficult, if not impossible to generate using Anymal’s current limb design. This observation highlights the need for an interactive design system like the one we describe in our work.

While the design of hybrid wheeled/legged robots demands specific considerations, so do the techniques used to control them. For traditional legged robots, motion planning frameworks are well-established. Generally speaking, existing techniques need to trade off computational efficiency against accuracy of the models that are used for motion planning. The FROST framework developed in [?], for example, accounts

for the full-body dynamics of a robot, but requires on the order of 5 to 10 minutes of computation to generate optimal motions. On the other end of the speed-accuracy spectrum, a state-of-the-art model predictive controller that requires about a millisecond of compute time was developed for the MIT cheetah robots [?]. To achieve this computational efficiency, a linearized model of the centroidal dynamics is used, and the robot’s root and end effector trajectories are computed using heuristics, rather than being treated as parameters in a trajectory optimization routine. The technique we use to generate motions for hybrid robots falls between these extremes. We use a centroidal model to capture the dynamics of the robots, and concurrently optimize for body and end effector trajectories, ground reaction forces and full-body poses, similar to the method described in [?]. Our trajectory optimization formulation is further augmented with various constraints that model the way in which different types of end effectors can move when in contact with the ground [1]. As we describe in this paper, when initialized with the solution of a novel warm-starting strategy, our trajectory optimization process converges in a matter of a few seconds, enabling an interactive exploration of the motion skills that different robot designs will possess.

Through a variety of sim-to-real experiments, we found that our motion optimization model is sufficiently powerful to generate a variety of dynamic maneuvers, including roll-walking while ducking under obstacles, roller-blading and ice-skating. Nevertheless, we note that for motor skills that are less dynamic in nature, motion planning techniques that consider only the kinematic design of a robot [8], [9], [10], [11], or approaches that use simplified dynamics model based on the zero moment point [12], [5] have also been proposed. Although potentially faster, these methods have not been shown to be capable of generating the types of motor skills

that we demonstrate in our work.

B. Contribution

In this paper, we build on the design and motion generation framework developed in [1] to create mobile robots of increased complexity. We demonstrate the benefits of our computational system as a predictive tool in the engineering of legged-wheeled robots by highlighting the important relationship between how a robot is designed and how it can locomote. As a first technical contribution, we present a numerical continuation method to efficiently warm-start the motion synthesis process for robot designs created with our interactive program. This warm-starting procedure drastically improves the robustness and convergence of the trajectory optimization routine employed by our framework. As a result, different robot morphologies can be tested very quickly in simulation, leading to an efficient exploration of the space of locomotion modes that different configurations of legged-wheeled robots are capable of. To validate our simulation results, we created two legged robots, shown in Fig. 1, that feature five actuated degrees of freedom per limb. Equipped with different types of end effectors - active or passive wheels and ice-skates - these robots are capable of an array of interesting locomotory behaviors that include walking, roll-walking, roller-skating and ice-skating. Our results suggest that a robot engineering system such as the one used in our work can become an invaluable tool in discovering robot designs that are specifically engineered for different tasks.

II. DESIGN AND MOTION GENERATION FRAMEWORK

Our interactive robot design framework builds on the computational system we developed in [1]. As illustrated in Fig. 2, robot designs are created interactively by selecting functional building blocks from a database and connecting them in a Lego-like manner. The database of available components is easily extendable, and our implementation includes servo motors, parameterized connectors that must be 3D printed, and different types of end effectors: active wheels whose angular speed is controlled by motors, passive wheels that spin freely around their wheel axis, and point, or circular-arc shaped feet.

User created robot designs serve as input for the trajectory optimization process that we use to automatically generate stable, physically-correct locomotion gaits. A motion plan editor allows users to also specify high-level commands, such as the footfall pattern to be used, desired movement speed, constraints on the body position or orientation at different moments in the locomotion cycle, etc. Our motion generation routine leverages a centroidal model of the robot's dynamics in addition to its kinematics model (Sec. II-A), and it outputs full-body motion trajectories that arise as solutions to a large system of non-linear equations. To improve the reliability and convergence rates of the underlying numerical solver, we introduce a novel warm-starting method (Sec. II-B) which makes it possible for designers to interactively explore the motion capabilities of vastly different robot morphologies in a matter of seconds. Motions generated with our framework can be verified either in a physically-simulated environment, or by having physical robots execute them directly.

A. Motion Plan Generation

The trajectory optimization formulation that we use in our framework was developed in [1]; we briefly summarize it in this section in order to set the stage for the novel warm-starting strategy that we introduce in this work.

Physically-valid motion plans that fulfill user-set motion targets are generated with trajectory optimization. Instead of using the dynamics of the whole body or a zero moment point (ZMP) approach, we utilize centroidal dynamics. It models the robot's linear and angular momenta using a single centroidal coordinate frame $\mathbf{c} = \{\mathbf{x}, \theta\}$, where \mathbf{x} is the robot's center of mass and θ the robot's global orientation. The robot interacts with its environment via its N end effectors by applying ground reaction forces $\mathbf{f} = \{\mathbf{f}^1, \dots, \mathbf{f}^N\}$ at the end effector's positions $\mathbf{e} = \{\mathbf{e}^1, \dots, \mathbf{e}^N\}$. For a planning horizon of hT , the motion plan is discretized in time using a direct transcription approach, where h is the length of a time step and T the number of time samples. For each time sample i , the Newton-Euler equations govern the relation between the centroidal coordinate frame \mathbf{c} and end effector forces \mathbf{f} and positions \mathbf{e} :

$$\sum_{j=1}^N \mathbf{f}_i^j + M\mathbf{g} = M\ddot{\mathbf{x}}, \forall i \quad (1)$$

$$\sum_{j=1}^N (\mathbf{e}_i^j - \mathbf{x}_i) \times \mathbf{f}_i^j = \mathbf{I}\ddot{\theta} + \dot{\theta} \times \mathbf{I}\dot{\theta}, \forall i \quad (2)$$

M and \mathbf{I} denote the total mass and inertia of the robot, and \mathbf{g} the gravitational acceleration.

In addition to the centroidal dynamics, constraints are introduced to ensure the physical validity of the end effector forces, such that the Coulomb friction model is satisfied:

$$\mathbf{f}_n \geq 0, |\mathbf{f}_t| \leq \mu |\mathbf{f}_n|$$

The motion plan must also respect a user-defined footfall pattern, which defines binary flags c for each time step [13]. If an end effector is in swing ($c = 0$), corresponding ground reaction forces \mathbf{f} need to vanish:

$$(1 - c)\mathbf{f} = 0$$

Each end effector is modeled as a wheel with wheel speed ω and rotation angles α that determine the wheel's orientation in the world frame. The end effector position \mathbf{e} of a wheel are defined as the contact point with the ground and $\rho(\alpha)$ is the vector connecting \mathbf{e} with the center of the wheel. Depending on the type of wheel, different constraints are introduced, whereas point feet are modeled by setting the wheel radius ρ to zero. The auxiliary variables describing the state of a wheel - ω , α and \mathbf{e} - allow us to formulate constraints that model the effect of wheels on the generated motion plan. For unactuated wheels, the ground reaction forces must vanish in the direction along which the wheel is free to move, which is given by the function $\mathbf{t}(\alpha)$:

$$\mathbf{f} \cdot \mathbf{t}(\alpha) = 0$$

End effector position trajectories \mathbf{e} of wheels that are in contact with the ground must be consistent with the wheel's orientation α and wheel speed ω . To ensure there is no slip between the

wheel and the ground, the velocity of the contact point of the wheel, given by $\boldsymbol{\omega} \times \boldsymbol{\rho}$, must be equal to the time derivative of the end effector position \mathbf{e} :

$$\left(\frac{\mathbf{e}_{i+1}^j - \mathbf{e}_i^j}{h} + \boldsymbol{\omega}_i^j \times \boldsymbol{\rho}(\alpha_i^j) \right) c_i^j = 0$$

The dynamics model is augmented with geometric constraints that ensure the centroidal coordinate frame and the end effector states are consistent with the robot's kinematic configuration. The robot's pose is described by minimal coordinates \mathbf{q} , which contains the root position and orientation as well as the joint angles. First, the center of mass \mathbf{x}_i and the robot's orientation θ must match the robot's root position and orientation given by the functions $\varphi_x(\mathbf{q}_i)$ and $\varphi_\theta(\mathbf{q}_i)$:

$$\varphi_x(\mathbf{q}_i) - \mathbf{x}_i = 0 \quad (3)$$

$$\varphi_\theta(\mathbf{q}_i) * R(\theta_i)^{-1} = I \quad (4)$$

Second, the wheel axis $\mathbf{a}(\alpha)$ defined by the wheel orientation α must be the same as the wheel axis computed from \mathbf{q} by the function $\varphi_b(\hat{\mathbf{a}}, \mathbf{q})$, with $\hat{\mathbf{a}}$ being the wheel axis in the body frame b of the rigid body the wheel is attached to:

$$\varphi_b(\hat{\mathbf{a}}^j, \mathbf{q}_i) - \mathbf{a}(\alpha_i^j) = 0 \quad (5)$$

And finally, the end effector position \mathbf{e} and wheel orientation α must be consistent with the position of the wheel's center of rotation, which is computed by the function $\varphi_b(\hat{\mathbf{l}}, \mathbf{q})$, where $\hat{\mathbf{l}}$ is the mounting location of the wheel in the body frame b :

$$\varphi_b(\hat{\mathbf{l}}^j, \mathbf{q}_i) + \boldsymbol{\rho}(\alpha_i^j) - \mathbf{e}_i^j = 0 \quad (6)$$

In addition, we introduce bound constraints on joint angles and velocities to account for motor limits.

All parameters at a time sample i are concatenated as the motion parameters

$$\mathbf{m}_i = \{\mathbf{q}_i, \mathbf{c}_i, \mathbf{e}_i, \mathbf{f}_i^1, \dots, \mathbf{f}_i^N, \boldsymbol{\omega}_i^1, \dots, \boldsymbol{\omega}_i^N, \alpha_i^1, \dots, \alpha_i^N\}$$

and the constraints as the vector $\mathbf{C}(\mathbf{m})$.

To give the user interactive control over the robot's motion, we formulate a set of objectives $O(\mathbf{m})$. For example, the user can ask for the robot's body to be at a target position \mathbf{x}_i^T at a specific moment in time i , which creates an objective $O_T(\mathbf{m}) = \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_i^T\|_2^2$.

Solving the resulting constrained minimization problem

$$\mathbf{m} = \operatorname{argmin}_{\mathbf{m}^*} O(\mathbf{m}^*) \text{ s.t. } \mathbf{C}(\mathbf{m}^*)$$

will produce a physically valid motion plan that fulfills the motion targets. We handle the constraints using a penalty based approach, and solve

$$\mathbf{m} = \operatorname{argmin}_{\mathbf{m}^*} E(\mathbf{m}) = \operatorname{argmin}_{\mathbf{m}^*} O(\mathbf{m}^*) + \frac{\mu}{2} \mathbf{C}(\mathbf{m}^*)^T \mathbf{C}(\mathbf{m}^*)$$

using Newton's method. For a complete description of the motion plan generation framework, we refer to [1].

B. Warm-Start Routine

The main goal of our computational framework is to enable an intuitive interactive approach to creating, evaluating and improving robot designs based on a desired set of locomotion skills. This is only possible if the entire design and motion generation pipeline is reliable and efficient. As summarized above, our framework generates full-body robot motions \mathbf{m} by finding a local minimum of $E(\mathbf{m})$, a function which encodes various motion objectives and constraints. This function is characterized by a highly non-linear and non-convex landscape, so it exhibits many undesirable local minima that standard gradient-based optimization methods are prone to getting stuck in. This challenge is particularly pronounced for motions that feature intermittent contacts with the environment, and the initial guess \mathbf{m}_0 that the optimization routine starts from plays a crucial role in determining which local minimum the final solution \mathbf{m} will converge to. In this section, we therefore turn our attention to formalizing a principled way of generating good initial solutions \mathbf{m}_0 for arbitrary robot morphologies that locomote using arbitrary gaits.

To gain intuition into the method we propose, consider the simple case of a quadrupedal robot that must locomote using a periodic walking gait. The motion of the robot's body must be appropriately synchronized with the corresponding footfall pattern; if the robot's center of mass is inadequately positioned at some point in the motion cycle, then as the robot will lift its next foot, it will lose balance and eventually fall. Such an initial guess for a motion plan corresponds to a very large objective value for the trajectory optimization objective $E(\mathbf{m})$, and it may require a very large number of iterations to be taken until a dynamically stable motion is found. In contrast, generating a motion where the robot simply stands in place is trivial. Our observation is that we can start from this trivial solution and formulate regularizing objectives that ask the robot to progressively unweight its feet according to the desired gait. As the robot does not have to lift its feet during this initial optimization process, it is able to maintain balance while figuring out how to coordinate the motion of its body with the underlying footfall pattern. Essentially, the robot first learns how to walk in place. We now show how to mathematically formalize this intuitive concept using a numerical continuation approach [14].

1) *Step 1*: To generate a favorable initial starting state \mathbf{m}_0 for the trajectory optimization routine described in the previous section, our warm-start method starts with a subset of the motion plan parameters and progressively introduces new degrees of freedom and constraints. As a first step, the footfall pattern and the corresponding constraints on the ground reaction forces are ignored. We then compute ground reaction forces that satisfy the centroidal dynamics constraints \mathbf{C}_d , given by Eqs. (1) and (2), for a quiet standing behavior:

$$\mathbf{f} = \operatorname{argmin}_{\mathbf{f}^*} \frac{1}{2} \mathbf{C}_d^T \mathbf{C}_d$$

2) *Step 2*: The next step in the warm-starting routine aims to synchronize the motion of the robot's body with the constraints imposed by the footfall pattern on the set of feasible ground reaction forces. In particular, during prescribed swing

TABLE I
OUTLINE OF WARM-START STRATEGY

step	motion param.	constraints and objectives	goal
1	\mathbf{f}	dynamics	robot stands on ground with all end effectors in stance
2	\mathbf{f}, \mathbf{c}	dynamics, zero \mathbf{f} in swing *	robot's body moves according to footfall pattern
3	\mathbf{q}, \mathbf{e}	kinematics, vertical end effector position trajectories *	robot poses that satisfy kinematics and swing trajectories
4	$\mathbf{q}, \mathbf{c}, \mathbf{e}, \mathbf{f}, \omega, \alpha$	all constraints, motion targets *	motion plan with full set of DOFs and achieving motion targets

* introduced iteratively

periods, end effectors must not apply any ground reaction forces. To introduce these constraints incrementally, we solve a sequence of N_1 optimization problems of the form:

$$\{\mathbf{f}, \mathbf{c}\} = \operatorname{argmin}_{\mathbf{f}_i, \mathbf{c}_i} \frac{1}{2} \mathbf{C}_d^T \mathbf{C}_d \text{ s.t. } (1 - c_i) \cdot \mathbf{f}_i^* < \frac{N_1 - k}{N_1 - 1} \mathbf{f}_i^1$$

where i denotes the index of end effectors, N_1 is the total number of iterations, $k \in \{0, N_1 - 1\}$ is the index of the current iteration, and \mathbf{f}^1 is the set of ground reaction forces computing in Step 1. At the start of this iterative process, the robot has the greatest control authority, as the end effectors that are in swing (i.e. $c = 0$) can still generate ground reaction forces that effect the robot's full-body motion. As k approaches N_1 , these forces are progressively driven towards 0, resulting in body trajectories that are smoothly optimized to anticipate the swing phases before they actually occur. We experimentally found $N_1 = 100$ to work reliably for a wide variety of robot designs and footfall patterns, and we use this setting for all our results.

3) *Step 3*: The intermediate motion plan generated in Step 2 satisfies the constraints imposed by the centroidal dynamics and the footfall pattern. Next, our goal is to find full body robot motions \mathbf{q} that match the trajectories of the body and end effectors in order to satisfy the kinematic consistency constraints \mathbf{C}_k , defined in Eqs. (3) – (6). To smooth out this optimization problem, we gradually increase the swing height from 0 to the user-specified value e_y^{des} and solve a sequence of sub-problems:

$$\{\mathbf{q}, \mathbf{e}\} = \operatorname{argmin}_{\mathbf{q}^*, \mathbf{e}^*} \frac{1}{2} \mathbf{C}_k^T \mathbf{C}_k + \frac{1}{2} \|e_y - \frac{k}{N_2 - 1} e_y^{\text{des}}\|_2^2$$

where N_2 is the number of iterations, $k \in \{0, N_2 - 1\}$ the current iteration, e_y the vertical component of the contact point and the desired swing height trajectory.

4) *Step 4*: The steps leading up to here have found a subset of motion parameters $\mathbf{f}, \mathbf{c}, \mathbf{q}, \mathbf{e}$ that satisfy the most challenging constraints. The final step in the warm-start routine now solves for the full set of motion parameters and includes all constraints. The values for all motion objectives (i.e. moving or turning speed) are set to 0 during this step, so the resulting motion corresponds to the robot locomoting in place. This motion constitutes the initial guess \mathbf{m}_0 that the trajectory optimization described in the previous section fine-tunes according to user-specified inputs.

III. ANALYSIS AND RESULTS

A. Evaluation of Warm-Start Routine

The goal of the warm-start strategy we introduced in section II-B is to appropriately initialize the non-linear program used

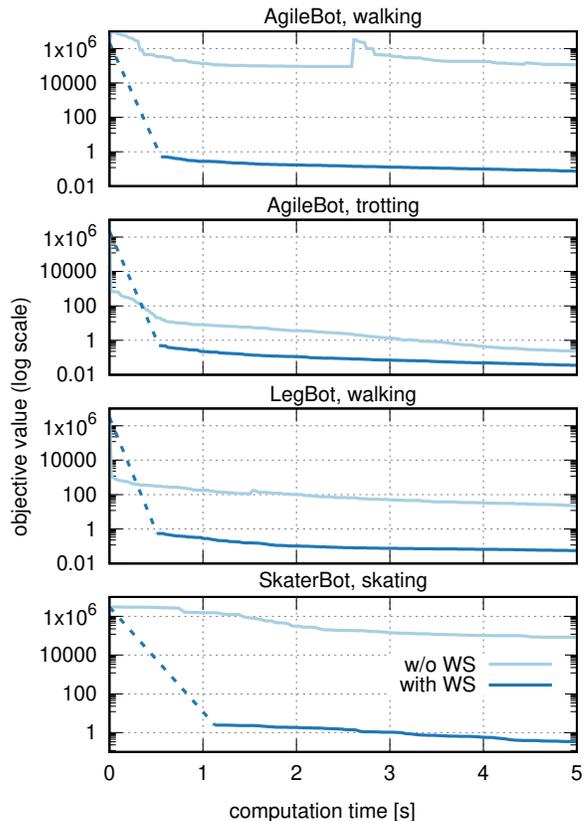


Fig. 3. Convergence rates for the trajectory optimization with and without the warm-starting procedure over 5s of computation time. From top to bottom: AgileBot (actuated wheels) employing a walking and trotting gait, LegBot (feet) walking, and SkaterBot (pairs of passive wheels) skating using a trotting footfall pattern. The computation time for the warm-starting procedure is illustrated through straight dotted lines. For all of these motion tasks, the robot was asked to move forward with a speed of 0.2m/s .

to generate optimal robot motions. As shown in Fig. 3, the convergence rates for the trajectory optimization process are greatly improved when using the initial solution obtained through warm-starting. We note that the trajectory optimization process can converge without warm-starting as well, but it is typically much slower. For AgileBot (quadruped robot, 16 joints, 4 actuated wheels) performing a trotting gait, for example, the trajectory optimization process takes about 9.5 seconds of compute time to reach the same objective value as the one obtained with warm-starting after just 5 seconds. However, the walking gait for the same robot fails to converge within 1 minute when warm-starting is omitted.

B. Morphology exploration for hybrid legged/wheeled robots

Traditional designs for legged robots with point feet are optimized to provide a sufficiently large reachable space for its end effectors. Adding passive or active wheels to a robot's feet demands further considerations in terms of the space of reachable end effector orientations, as this directly governs the functional capabilities of the final robot design. A robot whose wheels are actuated, for example, will need to move its limbs as it locomotes in a very different manner than one that propels itself using passive wheels or ice skates. Indeed, a robot design that is not compatible with a specific set of end effectors and desired motion skills may function poorly [7], or it may fail to locomote altogether.

Naturally, the question then arises: how does the morphological design of a robot shape its motor capabilities? We begin to study this question by using our computational framework to analyze a quadrupedal robot design that is equipped with one unactuated wheel per end effector – the SwizzleBot. In Table II we compare four different leg designs for SwizzleBot, with 2, 3, 4 and 5 motors per limb respectively. For each design we used our warm-starting and trajectory optimization routines to generate full-body motions using a desired speed of $0.3m/s$ as a target. All four robot designs successfully reached this target forward speed. However, the effort required to move as desired is vastly different. This can be seen in the last two columns of Table II which show the joint angle velocity and acceleration averaged over a motion cycle.

Indeed, the robot designed with five joints per limb stands out by being able to locomote with much smoother motion trajectories compared to the other three robot designs. To further clarify table II, we refer the interested readers to the accompanying video (<https://n.ethz.ch/~moritzge/ra-1-2020/>) for a side-by-side comparison of all four designs. We note that creating these designs and their corresponding motions

took only a matter of a few minutes with our computational framework.

C. From conceptual designs to physical prototypes

In order to test the results generated with our computational framework and motion optimization process, we built two quadrupedal robots that can be equipped with actuated or unactuated wheels as end effectors. Here, we give details on the fabrication process.

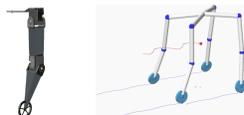
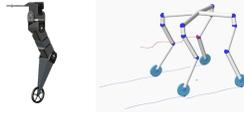
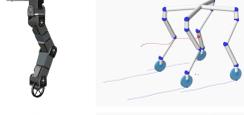
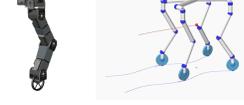
Both of our robots are made using 3d printed parts and off-the-shelf servomotors and electronic components. For 3D printing we used a Markforged Two machine that reinforces digitally manufactured parts with long strands of carbon-fiber. For actuation we used Dynamixel servomotors (XM430-W210T and XM540-W150T), both for the robot's joints and to power active wheels. An onboard Raspberry Pi 3B+ was used to send control signals to the servomotors. Fig. 5 shows some of the components we used to fabricate the physical robot prototypes. 3d-printing the structural components of the robot designs took about one week, while the assembly process took an additional half day.

Motion plans are generated offline on a desktop PC, sampled at 60Hz and then transferred to the Raspberry Pi wirelessly. A touch screen is mounted on the robot to provide an interactive interface that allows the robot operator to load new motion plans, combine multiple motions into a motion graph, and to set various servomotor parameters (e.g. the proportional and damping gains of the low-level PD controllers). We further use an Xbox gamepad controller to give high-level commands to the robot (i.e. which generated motion to execute). This intuitive and streamlined process makes it possible to try out the motions and behaviors generated with our framework in a matter of seconds.



Fig. 5. Our robots are made from 3d-printed parts and off-shelf electronic components for actuation and control. From left to right: 3D printed wheels that can be attached to a motor or left to spin freely, a roller blade with passive wheels and an ice skate, Dynamixel XM430-W210T are used to actuation joints and active wheels, and 3d-printed brackets.

TABLE II
DIFFERENCES IN SWIZZLEBOT DESIGNS AND MOTIONS

leg design	motion trajectories	avg. joint angle velocity [$\frac{\text{rad}}{\text{s}}$]	avg. joint angle accel. [$\frac{\text{rad}}{\text{s}^2}$]
2m		4.87	4408.22
3m		3.26	2572.32
4m		3.07	1870.08
5m		0.68	23.93

D. Robot Designs

a) *AgileBot - A Legged Robot with Actuated Wheels*: The first robot we designed is a quadruped that has four joints and one actuated wheel per limb. The joint degrees of freedom allow it to move sideways, turn in-place, move forward or sideways while ducking, and perform other agile motions; the accompanying video shows several motor skills where AgileBot makes efficient use of all its degrees of freedom. Note that all these motions emerge from the trajectory optimization process given only high-level targets as input. For example, in the *forward-duck* motion, we use a simple objective to ask the height of the robot's body to be below a user-specified threshold. Additional objectives can ask, for example, that

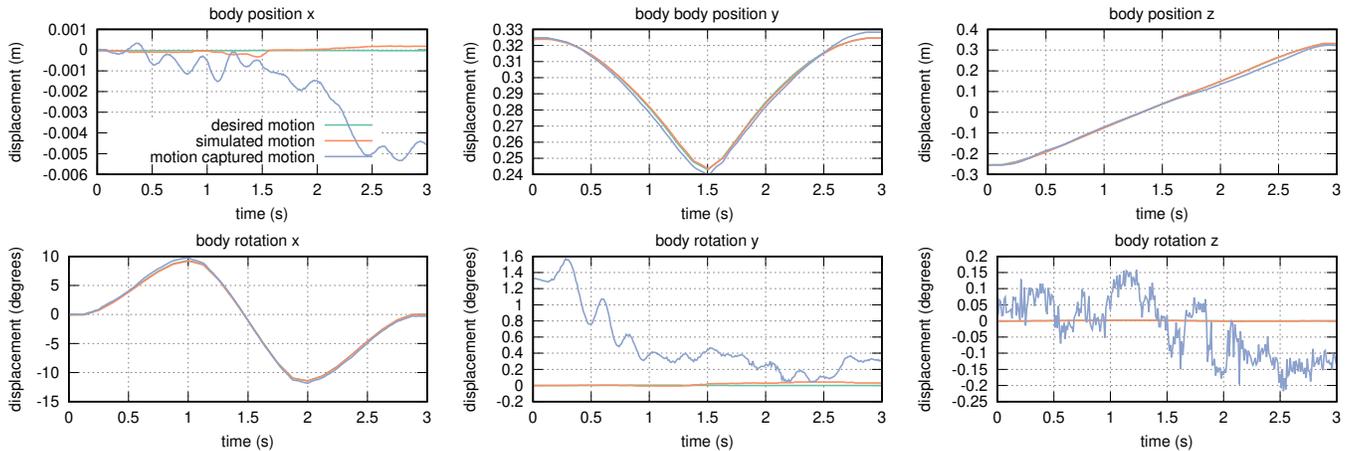


Fig. 4. Comparison of generated, physically simulated and motion-captured trajectories for AgileBot. The motor skill we evaluate here is the “forward-duck” motion, which can be seen in the accompanying video. The simulated and motion-captured trajectories match the motions generated through our optimization model very well. (Note the difference in scales of the vertical axes.)

the relative motion between the wheel and the body part it is attached to vanishes, essentially resulting in a design where the virtually welded wheel becomes a circular arc-shaped foot. We refer to this design as LegBot.

b) SkaterBot - A Legged Robot with Passive Wheels: As this design illustrates, equipping a legged robot with passive wheels can lead to fast, energetically favorable locomotion modes. To create SkaterBot, we used the same overall morphology as for AgileBot, but we exchanged the actuated wheels with roller blades – pairs of passive inline wheels. When both wheels are on the ground, the corresponding end effector can not rotate around the vertical axis without slipping. Skating motions therefore require the robot to lift its feet as it moves. To create an additional locomotion mode, the ankle joints can be rotated by 45 degrees, resulting in only one wheel of the roller blade being in contact with the ground. We call this design SwizzleBot, and we note that it is functionally equivalent to the ice skating robot that we also experimented with, given that the ice skates we employed have non-zero curvature. This robot design can propel itself without ever having to lift its feet.

E. Performance Evaluation

We used our trajectory optimization framework to generate a variety of locomotion modes for AgileBot, SkaterBot, LegBot and SwizzleBot. For each of the motions summarized in Table III, which can also be seen in the accompanying video, our goal was to maximize speed. For the AgileBot’s walking gait, we enforced the constraint that ensures the speed of the wheels is always zero, and we set the footfall pattern accordingly. For the rolling motion, the feet were set in stance mode for the entire duration of the gait cycle. For the walking and rolling behavior, the wheels were left free to rotate as best determined by the motion generator, and a walking footfall pattern was provided as input. We note that the speed limit of the wheels imposes strict constraints on the overall movement speed of the robot.

As our results show, the designs that feature passive wheels are much faster than AgileBot. This is particularly noteworthy because we used exactly the same types of servomotors to power the joints of the two robots. The skating motion, in particular, is more than twice as fast as AgileBot’s roll-walking locomotion mode.

Fig. 4 compares the motions generated by our trajectory optimizer against the motions executed by the physical robot. For this sim-to-real experiment, we measure differences in the motion of AgileBot’s body as it performs the “forward duck” maneuver that can be seen in the accompanying video. We use an OptiTrack motion capture system to record the motion trajectories executed by physical robot prototype. As can be seen, the planned position and orientation body trajectories for this relatively complex maneuver are tracked effectively both in simulation, and in the real.

TABLE III
MAXIMUM SPEED FOR DIFFERENT DESIGNS

Robot	measured max. speed
AgileBot walking	0.1483 m/s
AgileBot rolling	0.3380 m/s
AgileBot walking & rolling	0.4762 m/s
SwizzleBot	0.6897 m/s
SkaterBot	0.9677 m/s

IV. CONCLUSIONS AND FUTURE WORK

We introduced a novel warm-starting method that drastically improves the robustness and computational efficiency of the trajectory optimization routine that we use to generate optimal motions for legged robots. Integrated into our computational design framework [1], this warm-starting strategy promotes a systematic, interactive study of the ways in which the morphological design of a legged robot shapes its motor capabilities. To evaluate the predictions generated with our computational

framework, we created two physical robot prototypes that are capable of a rich array of motions and locomotory behaviors.

Our current efforts in establishing a computational workflow for robot engineering are not without limitations. In the future, for example, we plan to extend our computational design and motion generation framework to support robot morphologies that include mechanical transmission elements (e.g. linkages and drive belts), increasingly complex design features (e.g. kinematic loops that allow a robot's body to reshape as needed for different tasks), as well as soft materials (e.g. rubber feet, flexible links or springs used to store and release energy) that are appropriately accounted for within the trajectory optimization process. Furthermore, motion trajectories are currently played back in hardware in an open-loop fashion. While this suffices as an initial way of quantifying the success of sim-to-real transfer, robots operating in unstructured environments need the ability to adapt to unanticipated situations. This demands both advanced sensing capabilities, as well as the ability to adapt motion plans at faster-than-real-time rates based on incoming sensory inputs. Better numerical solvers and warm-starting procedures, full-body dynamics models, robust motion tracking controllers, as well as methods that can learn in an off-line process the output of trajectory optimization routines are all key to addressing this challenge; these topics will be the subject of our future investigations in this area.

REFERENCES

- [1] M. Geilinger, R. Poranne, R. Desai, B. Thomaszewski, and S. Coros, "Skaterbots: Optimization-based design and motion synthesis for robotic creatures with legs and wheels," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, p. 160, 2018.
- [2] S. Hirose and H. Takeuchi, "Study on roller-walk (basic characteristics and its control)," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3265–3270, IEEE, 1996.
- [3] J. A. Smith, I. Sharf, and M. Trentini, "Paw: a hybrid wheeled-leg robot," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 4043–4048, IEEE, 2006.
- [4] "Handle: Boston dynamics robot on wheels performs on stage. youtube. [online]. available: <https://www.youtube.com/watch?v=-7xvqqeo8c>,"
- [5] Y. de Viragh, M. Bjelonic, C. D. Bellicoso, F. Jenelten, and M. Hutter, "Trajectory optimization for wheeled-legged quadrupedal robots using linearized zmp constraints," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1633–1640, 2019.
- [6] M. Bjelonic, C. D. Bellicoso, Y. de Viragh, D. Sako, F. D. Tresoldi, F. Jenelten, and M. Hutter, "Keep rollin'—whole-body motion control and planning for wheeled quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2116–2123, 2019.
- [7] M. Bjelonic, C. D. Bellicoso, M. E. Tiryaki, and M. Hutter, "Skating with a force controlled quadrupedal robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7555–7561, IEEE, 2018.
- [8] W. Reid, F. J. Pérez-Grau, A. H. Göktoğan, and S. Sukkarieh, "Actively articulated suspension for a wheel-on-leg rover operating on a martian analog surface," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5596–5602, IEEE, 2016.
- [9] C. Grand, F. Benamar, and F. Plumet, "Motion kinematics analysis of wheeled-legged rover over 3d surface with posture adaptation," *Mechanism and Machine Theory*, vol. 45, no. 3, pp. 477–495, 2010.
- [10] T. Klamt and S. Behnke, "Anytime hybrid driving-stepping locomotion planning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4444–4451, IEEE, 2017.
- [11] M. Kamedula, N. Kashiri, and N. G. Tsagarakis, "On the kinematics of wheeled motion control of a hybrid wheeled-legged centaur robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2426–2433, IEEE, 2018.
- [12] A. Suzumura and Y. Fujimoto, "Real-time motion generation and control systems for high wheel-legged robot mobility," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 7, pp. 3648–3659, 2013.
- [13] V. Megaro, B. Thomaszewski, M. Nitti, O. Hilliges, M. Gross, and S. Coros, "Interactive design of 3d-printable robotic creatures," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, p. 216, 2015.
- [14] B. Krauskopf, H. M. Osinga, and J. Galán-Vioque, *Numerical continuation methods for dynamical systems*. Springer, 2007.