A Step Towards Generating Human-Like Walking Gait via Trajectory Optimization through Contact for a Bipedal Robot with One-Sided Springs on Toes

Kenneth Chao[1] and Pilwon Hur[1]

***Abstract*— Trajectory optimization with direct collocation are widely used in various bipedal walking studies, from dynamic simulation in biomechanics to efficient bipedal walking motion generation with multiple contact domains. Although the latter has gained popularity, most of the approaches in this field in general rely on pre-determined contact sequence (domains). This motivates us to use trajectory optimization through contact for generating an efficient and human-like walking gait, because this approach can automatically generate the contact sequence by solving a nonlinear program (NLP) with complementary constraints. However, in this approach the initial guess affects the result significantly, and the direct collocation with Euler method may not be accurate enough for the system dynamics. Therefore, we propose a modified framework and constraints to improve the generated results. We used a zero moment point (ZMP)-based flat-feet walking gait as an initial guess. We also show how to add virtual components like springs at ankle joints to alter the behavior of the resultant walking gait. In addition, considering the one-sided springs at the passive toe joints of the bipedal robot AMBER 3, additional complementary constraints are introduced for a better match of the full dynamics. The results of our modified approach with different constraint conditions are presented and discussed.**

I. INTRODUCTION

Generating dynamic walking gait of humanoid robots which is targeting on efficiency, agility and robustness is a challenging problem. For this motion generation problem, trajectory optimization is a powerful tool for solving the locally optimal trajectories for the dynamical systems which are potentially highly nonlinear. Among various methods in this field, trajectory optimization with direct collocation has gained more attention in recent years. In the field of biomechanics, the dynamic simulation using trajectory optimization is useful for studying neuromuscular coordination, predicting human behavior under various conditions [2], [4], or generating walking gait for lower-limb prosthesis control [14]. On the other hand, there are more and more applications using trajectory optimization for bipedal locomotion generation [10], [13], especially for bipedal robot controller design using Hybrid Zero Dynamics (HZD) scheme [7], [14], [6]. Although the trajectory optimization with direct collocation works well with HZD-based controller, this approach usually requires the specification of the contact sequence (or called domains/modes) as *a priori*. This specification can potentially make the problems much more complicated than needed since the combinatorics of the potential contacts need to be considered.

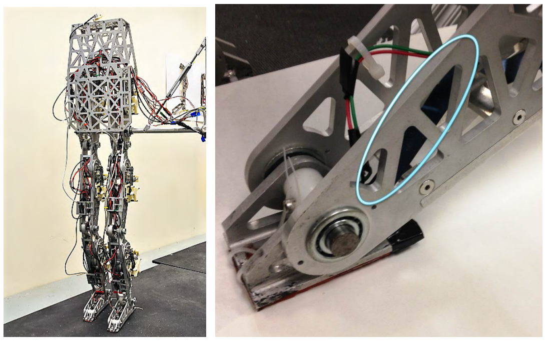


Fig. 1: The human-sized planar bipedal robot AMBER 3 (left). It is 148 cm tall, weights 33.4 kg, with 6 active degree of freedoms at hip, knee and ankle joints, capable of performing walking with multiple contact domains (e.g. walking with foot rolling motion). It has passive toes (right) with the torsional springs (circled by the bright blue loop).

On the other side of the spectrum, some researchers have developed approaches that do not require domain knowledge specific to the target behavior. Mortach et al. proposed the contact invariant optimization (CIO) [8] used for animations with simplified dynamic models. By introducing the contact-invariant cost and multiple optimization phases, this method optimizes over auxiliary decision variables which specify when and where the contacts are made, and can generate complex behaviors such as walking, climbing and handstand. Posa et al. [9] developed a unified framework termed *trajectory optimization through contact*, which has shown its capability to generate motion for high-dimensional systems with large number of modes, such as grasp planning, bipedal robot walking, or running. However, this local method will be affected largely by the choice of initial guess, and the accuracy of the numerical approximation using Euler method. In our work, for improving the accuracy and efficiency of the algorithm to generate walking gait under the similar optimization formulation, we first provide the modified framework and constraints for improving the numerical properties of the optimization formulation in Section III. Several schemes deigned for motion planning with better solution and better dynamical system description for the passive toes of the bipedal robot AMBER 3 (Fig. 1) are presented in Section IV. Results and conclusions are presented in Section V and Section VI respectively.

[1] Kenneth Y. Chao and Pilwon Hur are with the Department of Mechanical Engineering, Texas A&M University, 3123 TAMU, College Station, TX 77843, USA. {kchao,pilwonhur}@tamu.edu

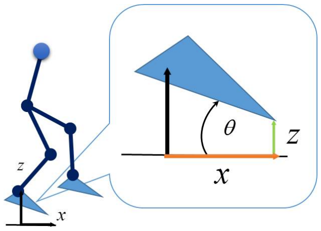


Fig. 2: The schematic of a bipedal robot with a floating base.

II. Full Dynamics and Bipedal Locomotion

A. System Dynamics with Contact Constraints

The dynamics of a rigid body model with a floating base can be expressed as follows:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = Bu + J^T\lambda \tag{1}$$

where q is the generalized coordinate that includes $[x, z, \theta]^T$ (Fig. 2), $D(q)$ is the inertia matrix, $C(q,\dot{q})$ is the Coriolis matrix, $G(q)$ is the gravity vector, J is the Jacobian matrix of the contact position $\phi(q)$ such that $J = \partial\phi/\partial q$, B is the torque distribution matrix, u is the control input, and λ is the contact force. Any potential contact point of the system can be described as $\phi(q) = [\phi_x(q), \phi_z(q)]^T$. $\phi_x(q)$ is tangential to the contact surface and $\phi_z(q)$ is the normal distance. A contact is made when $\phi_z(q)$ reaches zero.

For each potential contact point $\phi(q)$ with the contact force $\lambda = [\lambda_x, \lambda_z]$ and velocity $\gamma = J\dot{q} \triangleq [\gamma_x, \gamma_z]$, if the sliding contact is permitted, a set of *complementary constraints* [9] can be used to describe the Coulomb friction model:

$$\phi_z(q),\ \lambda_z,\ |\gamma_x| \geq 0 \tag{2}$$

$$\mu\lambda_z - |\lambda_x| \geq 0 \tag{3}$$

$$\phi_z(q)\lambda_z = 0 \tag{4}$$

$$(\mu\lambda_z - |\lambda_x|)|\gamma_x| = 0 \tag{5}$$

where μ is the friction coefficient. On the other hand, if sliding is not allowed, instead of eq. (5), the constraint eq. (6) needs to be satisfied:

$$\lambda_z|\gamma_x| = 0 \tag{6}$$

B. Domains of Bipedal Robot Walking

In our setup, the model of AMBER 3 has four potential contacts: the heels and toes on both feet. The contact conditions for each foot can be defined as: i) toe-off ($\phi_{z,toe} = 0,\ \phi_{z,heel} > 0$), ii) heel-contact ($\phi_{z,toe} > 0,\ \phi_{z,heel} = 0$), and iii) flat contact ($\phi_{z,toe} = 0,\ \phi_{z,heel} = 0$). A specific contact domain in the bipedal robot walking gait will be determined by the contact conditions of both feet.

C. Trajectory Optimization and Locomotion Generation for a System with Multiple Domains

A trajectory for a dynamic system can be treated as a set of state as a function of time $x(t)$ resulting from its initial condition $x(t_0)$ and control $u(t)$. As the name implies, *trajectory optimization* is a set of local methods for planning the optimal trajectory $x(t)$ with $u(t)$, and minimizing the objective cost over a horizon $t \in [0, T]$.

Among various approaches, *trajectory optimization with direct collocation* in general has nicer numerical properties than the *indirect method*. Thus, it can be used to solve complex problems such as a system with multiple contact domains. For handling this type of problems, most state-of-the-art techniques assume the contact sequence is known or specified, and then the contact condition is either inserted explicitly through constrained dynamics, or implicitly through dynamics constraints like eq. (1). This approach works well with HZD schemes, where a set of boundary conditions between domains (especially the ones with impact maps based on inelastic collision) need to be satisfied [7], [14], [6]. However, several potential issues may arise with the increase of contact points, such as determining the optimal contact sequence, and the increasing restriction for searching trajectories with more and more impact maps, which may rule out some potentially feasible trajectories. Therefore, the *optimization through contact*, which treats the contact sequence as a part of the trajectory becomes a nice resolution for those issues, will be introduced in next section.

III. Trajectory Optimization through Contact with Direct Collocation

Inspired by the time stepping method [11] used for forward simulation, Posa et al. [9] proposed the *trajectory optimization through contact*. The main idea of the time stepping method is to discretize the system state and control to formulate the multi-contact dynamics as a Linear Complementarity Problem (LCP). In this way, only the contact force acting over a period will be considered, which eliminates the need to differentiate between continuous and impulsive forces. Similarly, using direct collocation with trapezoid method, the optimization through contact directly optimizes the given cost function over a set of free variables including feasible states, control inputs, contact forces, time step and other slack variables. The general trajectory optimization through contact can be stated as:

$$\underset{\mathbf{x}=[h,x_1,\ldots,x_N,u_1,\ldots,u_N,\lambda_1,\ldots,\lambda_N]}{\operatorname{argmin}} \quad g_f(x_N) + h\sum_{k=1}^{N} g(x_k, u_k) \tag{7}$$

$$\text{s.t.} \quad \mathbf{x_{min}} \leq \mathbf{x} \leq \mathbf{x_{max}}$$

$$\mathbf{f_{min}} \leq \mathbf{f}(\mathbf{x}) \leq \mathbf{f_{max}}$$

where h is the time step, x_k is the discretized state variables $[q_k; \dot{q}_k]^T$ at k^{th} time step, $g_f()$ represents the final cost, and $h\sum g()$ is the integral cost. $\{\mathbf{x_{min}}, \mathbf{x_{max}}\}$ and $\{\mathbf{f_{min}}, \mathbf{f_{max}}\}$ are the vectors corresponding to the lower and upper bounds of decision variables and constraints respectively. Though the direct collocation scheme largely increases the number of free variables and constraints, the well-posed nature of the problem as a large sparse Non-Linear optimization Problem (NLP) (with sparse Jacobian matrices of the cost function and constraints) allows nonlinear optimization solvers like IPOPT [12], [6] and SNOPT [5], [9] to solve the NLP efficiently.

However, despite the concise and unified framework that can automatically derive the contact sequence by solving the NLP in eq. (7), there exist several issues. For example, general walking motion generation usually requires 20 to 50 collocation points for one half gait cycle (i.e. a single step), where the step size is in the order of 10^{-2} second. However, the step size required for time stepping method with more accurate simulation result is in millisecond [11]. Therefore, the time step size h may not be small enough for accurate dynamic simulation. As a result, a transcription method with higher accuracy, such as Runge-Kutta method or Hermite-Simpson method should be considered to decrease the integral error. In addition, the initial guess to this local method [9] and the related scheme for relaxation of the complementary constraints also need to be carefully handled for improving the optimization performance and the quality of the generated gait. In this section, we will focus on the transcription using Hermite-Simpson method with the corresponding constraint setup, followed by the introduction of the cost function and other important constraints in our modified optimization framework. The related constraint relaxation scheme and other adjustments will be introduced in the next section.

A. General Setup

For the discretization of all state variables, we set the time step as $h = T/N$, where $T > 0$ is the duration of a half gait cycle and $N = (2N_c + 1)$ is the number of collocation nodes. To use the Hermite-Simpson method to describe the relationship between the adjacent state variables, the number of *cardinal nodes* N_c needs to be selected first. In this case, the odd points $(x_1, x_3, \ldots, x_{2N_c+1})$ are the cardinal nodes where the time duration between any adjacent cardinal points can be arbitrarily chosen. For simplicity, the fixed time step h is used in our framework, so the duration between two adjacent cardinal points is $2h$. On the other hand, the even points called interior points $(x_2, x_4, \ldots, x_{2N_c})$ need to be placed at the center of two adjacent cardinal nodes.

B. Transcription for Direct Collocation Using Hermite-Simpson Method

In our modified framework, the Hermite-Simpson method is chosen for improving the accuracy of numerical approximation for kinematics and dynamics [6], [10]. The constraints of this method can be expressed as:

$$x_k - \frac{1}{2}(x_{k+1} + x_{k-1}) - \frac{1}{8}h(\dot{x}_{k-1} - \dot{x}_{k+1}) = 0 \qquad (8)$$

$$x_{k+1} - x_{k-1} - \frac{1}{6}h(\dot{x}_{k-1} + 4\dot{x}_k + \dot{x}_{k+1}) = 0 \qquad (9)$$

The physical meaning of the constraints above is that the k_{th} state variable and its time derivative x_k and $\dot{x}_k$ on the interior node (approximated as a cubic spline) should match the state variables and its time derivative evaluated through the system's kinematic and dynamic equations explicitly. Note for this Hermite-Simpson method with local compression [3], only the states of cardinal nodes belong to the free variables, since the state and its derivative of the interior node can be explicitly calculated based on the constraints stated above. This method is referred as *the compressed form of Hermite-Simpson method* [3].

C. The Implicit Constraint Expression with Extra NLP Variables

As per the previously mentioned explicit calculation (the acceleration $\ddot{q}_i = f(q_i, \dot{q}_i)$ from the system dynamics), the state on the interior node in eqs. (8) and (9) is coupled with the states on adjacent grid points. In addition, this calculation requires the inverse of the inertia matrix. Though the compressed form of Hermite-Simpson method using less decision variables (because the states of internal nodes are functions of cardinal nodes), it can limit the sparsity of the Jacobian matrix of constraints. To improve the sparsity, extra NLP variables, such as $\ddot{q}_k$ of the cardinal points and the interior points' states and accelerations (x_i and $\ddot{q}_i$) is introduced so that the set of free variables in eq. (7) becomes:

$$\mathbf{x} = [h, x_1, \ldots, x_N, \ddot{q}_1, \ldots, \ddot{q}_N, u_1, \ldots, u_N, \lambda_1, \ldots, \lambda_N]$$

On the other hand, instead of explicitly calculating the $\ddot{q}_k$ for all nodes with the inverse inertial matrix, the dynamic constraints in eq. (1) for each time step are inserted. Please refer to [3] for *Hermite-Simpson (Separated) method (HSS)* in Chapter 4 for further details and discussions.

D. Cost Functions and Constraints

Similar to the previous works using trajectory optimization with direct collocation [6], [9], [10], the mechanical cost of transport (COT) is used with an additional sum of torque squared with a small scaler factor ω as shown:

$$\mathit{cost}(\mathbf{x}) = \frac{1}{mgd}\sum_{k=1}^{N}\sum_{i}|u_{k,i}\dot{q}_{k,i}| + \omega\sum_{k=1}^{N}u_k^T u_k \qquad (10)$$

where mg is the system total weight, and d is the total traveling distance. Empirically, the sum of torque squared can help to improve the generated gait. Except for the kinematic and dynamic constraints (eqs. (1), (8) and (9)), other important constraints are summarized here:
Contact Constraint. Depending on whether the sliding contact is allowed or not, either eqs. (2) to (5) (referred as SACC: sliding allowed contact constraints) or eqs. (2) to (4) and (6) (referred as NSCC: non-sliding contact constraints) need to be satisfied for all cardinal and interior points.
Periodic Constraints. To generate the nominal walking gait, the periodic constraints are expressed as shown:

$$q_1 - Rq_N = 0 \qquad (11)$$

$$\dot{q}_1 - R\dot{q}_N = 0 \qquad (12)$$

$$\dot{x}_{1,COM} = \dot{x}_{N,COM} \qquad (13)$$

$$z_{1,COM} = z_{N,COM} \qquad (14)$$

$$\dot{z}_{1,COM} = \dot{z}_{N,COM} \qquad (15)$$

$$x_{N,COM} \geq x_{1,COM} + d_{min} \qquad (16)$$

where R is the relabeling matrix which switches the state variables at the joints on the left leg to the right or vice versa, and d_{min} is the minimum moving distance of x_{COM}.

Contact Constraint of Stance Toe. Since the duration of toe contact is slightly longer (about 55% for a full gait cycle) than one half gait cycle in the human walking gait analysis, the following constraint can further simplify the optimization without altering the objective for human-like walking gait generation, i.e. the stance toe ($stoe$) is constrained as:

$$\phi_{stoe}(q_k) = 0 \tag{17}$$

for $k = [1, \ldots, N]$. This constraint can also help to eliminate some undesirable gaits, i.e. the walking motion which includes the hopping in the half gait cycle.

IV. Running the Optimization Towards Generating Human-like Walking Gait

With the modified framework of the trajectory optimization through contact, we improve the numerical approximation accuracy, the sparsity of the Jacobian matrix about kinematic and dynamic constraints. In addition, the quality of the generated gait is improved by introducing the additional terms in cost function. However, sometimes it is still tricky to derive a high-quality gait by solving the NLP just a few times. There are several potential reasons for that. First of all, the choice of the initial guess can lead to different feasible gaits that satisfy all the constraints as mentioned in [9]. Naturally, more human-like initial guesses or the cost function for fitting human data may lead to a more desirable result. But, such approaches would be deliberatively guiding the optimization towards human-like gait. Our objective, on the other hand, is to naturally generate a human-like gait through trajectory optimization with general constraints and initial guess that is easy to generate. We believe that such an approach would be applicable for prosthesis, orthosis, or exoskeletons, because the produced results would be favorable to both the robotic systems and the humans interacting with them. Second, dependent on sliding contact condition, the generated behavior from the same initial condition may vary a lot because the different contact constraints have different numerical properties. Last but not least, since humans inherently have more passive components compared to a pure rigid-body model, it may be helpful to introduce some virtual components to slightly alter the generated gait. In the following subsections, the series of adjustments and schemes for improving the generated gait will be introduced.

A. Choice of the Initial Guess

As per our objective, the ZMP-based flat walking gait is chosen as the initial guess. The reasons for this choice are: i) The ZMP-based method for walking motion generation is widely-used. ii) With the simple flat-contact condition and two domains (i.e. the single support and double support phases), it is relatively easy to derive a dynamically feasible trajectory using constrained dynamics. iii) The generation of desired trajectory for ZMP-based walking (e.g. the ZMP trajectory, end-effector trajectory, torso angle) is straightforward.

B. Choice of Contact Constraints

In our implementation with the ZMP-based walking gait as the initial guess, the generated gait with SACC and NSCC are quite different, as shown in the next section. For the gait with SACC, the foot clearance of the swing foot is relatively small; the foot is almost sliding along the ground until it makes a step. On the other hand, the gait with NSCC behave more like a passive walker, which has a slightly larger sway-up motion before the heel-strike.

C. Virtual Springs on Ankles for Inducing Heel-strike Motion

For the resulting gaits that have no obvious heel-strike even with a larger d_{min} in eq. (16), one potential solution is to add virtual passive components to the system. Here we choose to add a torsional spring with a small stiffness k to the ankle joint to emulate the effect of the human Achilles tendon at the ankle (which prevents the foot from dropping even when the ankle is relaxed). The equation of motion then becomes:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = Bu + J^T\lambda - kBB_kq \tag{18}$$

where BB_k is the spring torque distribution matrix for assigning the spring torques to the torque equations of ankle joints. With this setup, the resultant torque applied to the system becomes $u - kB_kq$. For mitigating the effect of introducing virtual elastic components, the second term in the cost function eq. (10) can be modified as follows:

$$\omega \sum_{k=1}^{N} (u_k - kB_kq_k)^T(u_k - kB_kq_k) \tag{19}$$

D. Contact Constraints for One-sided Springs on Toes

For the bipedal robot AMBER 3, a set of torsional springs are attached on the passive toe joints (Fig. 1). The mechanical joint limit is designed that the torsional spring will activate only when the foot is in toe-off condition. Therefore it can be approximated as a contact point which has a one-sided torsional spring. An additional set of complementary constraints for the toe with one-sided spring then can be expressed as follows:

$$k_{toe}\theta_{toe} = (T_1 + T_2) - T^- \tag{20}$$

$$(T_1 + T_2)T^- = 0 \tag{21}$$

$$T_1T_2 = 0 \tag{22}$$

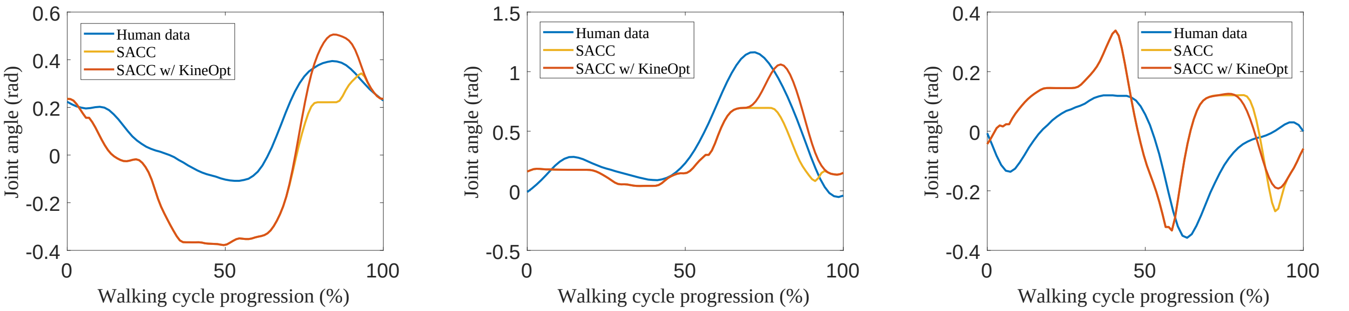
$$\phi_{z,toe}T_1 = 0 \tag{23}$$

$$\phi_{z,toe}, T_1, T_2, T^- \geq 0 \tag{24}$$

where the k_{toe} is the stiffness, T_1, T_2, and T^- are slack variables for the one-sided spring, where the real torque applied to the system through the active spring is the variable T_1. Under this setup, the equation of motion becomes:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = Bu + J^T\lambda - J_{\theta,toe}^T T_1 \tag{25}$$

where the $J_{\theta,toe}$ is the Jacobian matrix of the toe orientation.



(a) The trajectories of the hip joint.

(b) The trajectories of the knee joint.

(c) The trajectories of the ankle joint.

Fig. 3: The angular trajectory comparison between human data, gait SACC and gait SACC with the kinematic optimization.

E. Relaxations on the Complementary Constraints

For solving this problem using SNOPT with sparse sequential quadratic programming, it has been reported that it is practically useful to temporarily relax the complementary constraints [9] as follows:

$$M(x),\ N(x) \geq 0 \tag{26}$$

$$M(x)N(x) = \epsilon \tag{27}$$

where ϵ is a small nonnegative constant. On the other hand, we also found when using IPOPT (based on a primal-dual interior point method) to solve this type of the problem, the relaxation is also required. Without the relaxation, the primal-dual barrier approach will drive $M(x)$, $N(x)$ away from the boundary and leads to the worse local solution and convergence of the optimization problem. Since ϵ is sensitive to the optimization problem, we empirically used a simple grid search in the range of 10^{-3} to 10^{-1}, with a smaller maximum iteration number of the solver for quickly choosing a ϵ for a better start.

F. A Kinematic-based Trajectory Optimization for Increasing the Foot Clearance

Another observation from the result of the optimization through contact is that the ground clearance of the swing foot can be very small, probably caused by the minimization of the objective function which contains cost of transport. In general it should be improved by inserting the contact constraint in the form of $\phi_z(x) \geq f(x)$, in addition to the constraint in eq. (2). However, practically it might easily be compromised by the relaxation of complementary constraints. For increasing the foot clearance effectively with minimal effect on the original gait, a kinematic-based trajectory optimization for the swing leg trajectory is adopted as a post processing:

$$\begin{aligned} \underset{\mathbf{x}=[x_1,\ldots,x_N]_{swing}}{\text{argmin}} \quad & \sum (q_k - q_{k,ref})^T (q_k - q_{k,ref}) \\ \text{s.t.} \quad & \phi_{z,toe} \geq f(\phi_{x,toe}) \\ & \phi_{z,heel} \geq f(\phi_{x,heel}) \\ & \text{kinematic constraints in eqs. (8) and (9)} \end{aligned} \tag{28}$$

where $q_{k,ref}$ is the joint trajectory derived from optimization through contact, and $f(\phi_x)$ is a normal distribution function of the contact point's horizontal position.

V. Optimization Results and Related Comparisons

The formulated optimization with different constraints were solved using IPOPT with the linear solver ma57. Depending on the relaxation and the initial guess, the required computation time varied from 30 seconds to 10 minutes. For the common parameter setup applied for all the cases, $\omega = 10^{-3}$, $k = 10Nm/rad$, $d_{min} = 0.5m$. Except for the ZMP-based walking (ZMP) as the initial guess, other generated gaits for comparison include: the optimization using SACC (SACC), the optimization using NSCC (NSCC), and the optimization with one-sided spring constraints and NSCC (OSS) (as shown in Fig 4 to Fig. 6). The main quantities for comparison are listed in TABLE I.

TABLE I: The list of the modified costs, stride lengths and double support percentage values for initial guess, and generated gaits with different contact constraints.

Gait type	ZMP	SACC	NSCC	OSS
Cost	0.577	0.048	0.049	2.664
Stride length	0.2m	1.10m	1.0m	1.0m
Double support percentage	33.33%	31.37%	35.48%	35.48%

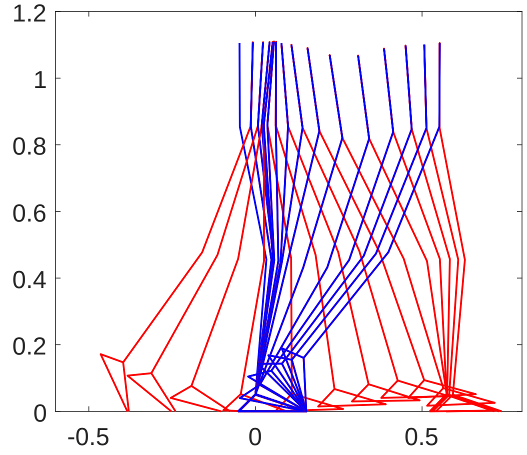


Fig. 4: The walking tile of the generated gait with SACC.

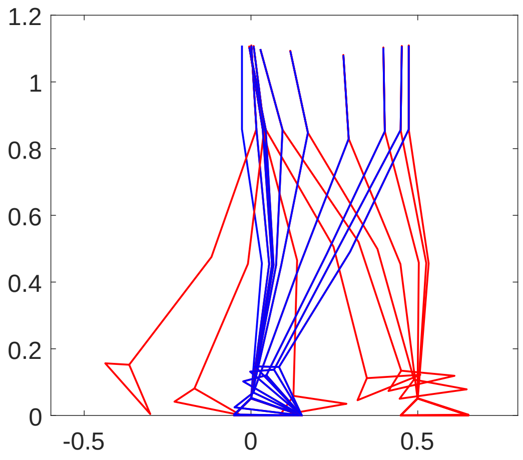


Fig. 5: The walking tile of the generated with NSCC.

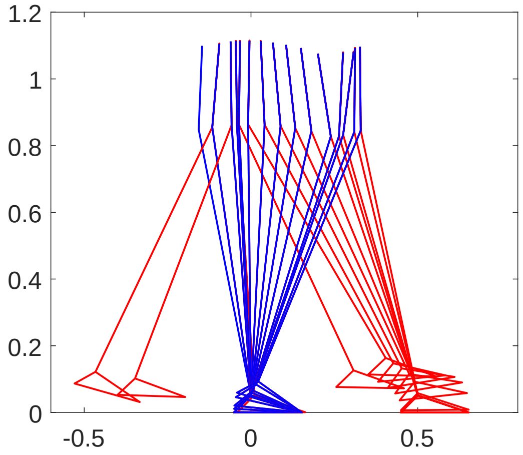


Fig. 6: The walking tile of the generated gait with OSS.

The generated gaits with NSCC and SACC show that these two types of constraints can generate different gait characteristics. Thus the user can interchangeably switch among different constraints during the optimization process for getting a better result. In Fig 3, the human gait, the gait with SACC before and after the kinematic optimization for increasing the swing foot clearance are compared. Although the discrepancies still exist, patterns of the gait with SACC and kinematic optimization are closer to the human ones. The differences observed in the ankle trajectories are larger than those of the knee and hip, but the concluding stage of the ankle trajectory with SACC and kinematic optimization is similar to the initial stage of the human ankle trajectory. Further adjustments of the introduced schemes are required to improve the phase difference here.

For the gait with one-sided spring constraints, although the constraints helps to decrease the toe-off angle (Fig. 6), the minimum cost was still quite high compared with the other gaits since there were more complementary constraints need to be satisfied or relaxed at the same time.

VI. CONCLUSIONS AND FUTURE WORK

With the modified framework, a series of gaits with different constraints are generated. To make the optimization with complementary constraints more tractable (e.g. for one-sided spring), other solvers using SQP method should also be considered. Further adjustment of the parameters for the provided schemes is also required for more natural gait generation. On the other hand, further validations including more simulations and physical implementations are still required for testing the stability and robustness of the gait. Currently, the testing experiment is being undergone [1], and the artifacts from the support mechanism and treadmill will be resolved. We also plan to use this method for generating the trajectories for lower-limb prosthesis and exoskeleton.

ACKNOWLEDGMENT

The authors thank Namita Anil Kumar and Kenny Chour for their assistance on this work.

REFERENCES

[1] Experiment of walking with multiple domains on AMBER 3. `https://youtu.be/4QS9QBgkGrQ`.

[2] M. Ackermann and A. J. van den Bogert. Optimality principles for model-based prediction of human gait. *Journal of Biomechanics*, 43(6):1055 – 1060, 2010.

[3] J. T. Betts. *Practical methods for optimal control using nonlinear programming*. Advances in design and control. Society for Industrial and Applied Mathematics, 2010.

[4] F. De Groote, A. L. Kinney, A. V. Rao, and B. J. Fregly. Evaluation of direct collocation optimal control problem formulations for solving the muscle redundancy problem. *Annals of Biomedical Engineering*, 44(10):2922–2936, 2016.

[5] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):99–131, 2005.

[6] A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. D. Ames. 3d dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1447–1454, May 2016.

[7] A. Hereid, S. Kolathaya, and A. D. Ames. Online optimal gait generation for bipedal walking robots using legendre pseudospectral optimization. In *IEEE Conference on Decision and Control (CDC)*, pages 6173–6179, Dec 2016.

[8] I. Mordatch, E. Todorov, and Z. Popović. Discovery of complex behaviors through contact-invariant optimization. *ACM Transactions on Graphics (TOG)*, 31(4):43:1–43:8, July 2012.

[9] M. Posa, C. Cantu, and R. Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *The International Journal of Robotics Research*, 33(1):69–81, 2014.

[10] M. Posa, S. Kuindersma, and R. Tedrake. Optimization and stabilization of trajectories for constrained dynamical systems. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1366–1373, May 2016.

[11] D. Stewart and J. C. Trinkle. An implicit time-stepping scheme for rigid body dynamics with coulomb friction. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 162–169, 2000.

[12] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[13] W. Xi and C. D. Remy. Optimal gaits and motions for legged robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3259–3265, Sept 2014.

[14] H. Zhao, A. Hereid, E. Ambrose, and A. D. Ames. 3d multi-contact gait design for prostheses: Hybrid system models, virtual constraints and two-step direct collocation. In *IEEE Conference on Decision and Control (CDC)*, pages 3668–3674, Dec 2016.