

Animated Cassie: A Dynamic Relatable Robotic Character

Zhongyu Li, Christine Cummings and Koushil Sreenath

Abstract—Creating robots with emotional personalities will transform the usability of robots in the real-world. As previous emotive social robots are mostly based on statically stable robots whose mobility is limited, this paper develops an animation to real-world pipeline that enables dynamic bipedal robots that can twist, wiggle, and walk to behave with emotions. First, an animation method is introduced to design emotive motions for the virtual robot’s character. Second, a dynamics optimizer is used to convert the animated motion to dynamically feasible motion. Third, real-time standing and walking controllers and an automaton are developed to bring the virtual character to life. This framework is deployed on a bipedal robot Cassie and validated in experiments. To the best of our knowledge, this paper is one of the first to present an animatronic dynamic legged robot that is able to perform motions with desired emotional attributes. We term robots that use dynamic motions to convey emotions as *Dynamic Relatable Robotic Characters*.

I. INTRODUCTION AND RELATED WORK

Pixar characters are full of life and are emotive. However, they are virtual characters and only live in our screens. Robots living in real world are inanimate characters designed to autonomously execute tasks. For robots to meaningfully interact with humans, they need to go beyond task execution and incite emotion in humans through their actions. Robots need to be *relatable*. Most state-of-the-art social robots are statically stable [1], *i.e.*, they are either small mobile robots [2], or person-size ones with heavy bases [3], [4], or lack mobility [5], [6]. Their capacity to convey emotion is limited to simplistic visual cues, like facial expressions, and cannot use body language to interact with humans. Conversely, dynamically stable robots, specifically bipedal robots, can move and convey emotion with much more agility and success. Unfortunately, as legged robots are high-dimensional, nonlinear and underactuated systems with hybrid dynamics, this kind of human-robot interaction has not been explored.

In this paper we seek to ascertain the feasibility of utilizing a dynamic bipedal robot, Cassie Cal, to perform natural emphatic motions which ascribe a personality to a robot and enable transcending from being a “robot”, as shown in Fig. 1. Cassie is a bipedal walking robot that is about one-meter high while standing and has 20 Degrees-of-Freedom (DoF) among which are 5 actuated motors and 2 passive joints connected by leaf springs on each leg.

To address the needs for creating dynamic relatable robotic characters, three sub-problems need to be solved:

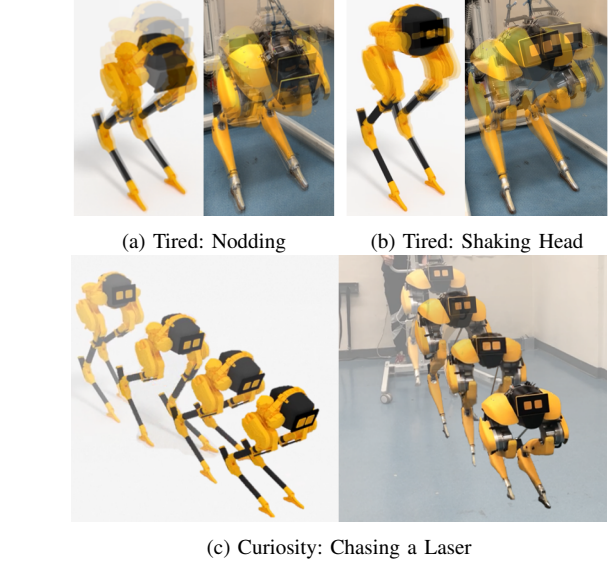


Fig. 1: Cassie’s emotive motions: the robot’s emotive behaviors are animated (left) and our proposed framework enables the dynamic robot Cassie to perform the designed motions in real time (right). Experimental video is at <https://youtu.be/zjzfzfgb3oK8>.

- Prob. 1** Provide an animation tool to design behaviors with emotional attributes for robots. Typically, such motions are kinematic and may not be feasible for dynamic robots.
- Prob. 2** Ensure the generated motions are dynamically-feasible for the robot while staying as close as possible to the original animated motions. This could be an optimization problem.
- Prob. 3** Develop a real-time controller to enable the dynamic robot to perform the optimized motions while balancing.

Effective solutions to the above-mentioned problems provide a tool to bring emotive motions to dynamic robots in real life.

A. Related Work

For Prob. 1, algorithmic robot motion planning [7] is designed to execute certain tasks with given objective functions, thus resulting in inanimate motions. To generate emotive motion for robots, there are very few solutions. *Animation creation*, as is commonly used in the entertainment industry, is one feasible approach. iCAT [8], followed by andyRobot [9], Cozmo [2], and Disney’s Vibration-Minized Animated Character [10], developed animation engines to enable the robots to combine multiple interactive behaviors with believable robot animations. Animation is a powerful tool for robotic behavior design, as animators can draw and modify robot motions interactively with visual effects [11], [12]. However, almost all previous efforts to bridge animation and robotics has been on statically stable robot platforms with little attention focused on the dynamic properties of

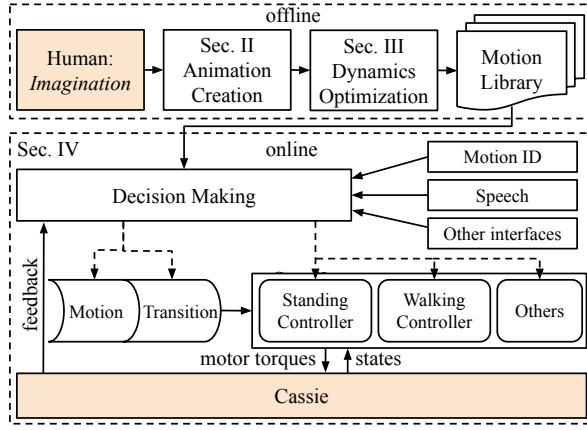


Fig. 2: Framework for the dynamic relatable robotic character Cassie to behave animatedly. This pipeline enables humans to create virtual character behaviors from their imagination. After dynamics optimization, the animated motions are stored in a motion library and can later be performed by Cassie in real life.

legged robots. One approach widely used in humanoid-robot fields [13]–[15], is imitating human motion. This method provides a tool to generate robot motion by imitating human motions, thereby addressing Prob. 1. Prob. 2 is solved by constructing a Zero-Moment-Point (ZMP)-based optimization [16] for the humanoid robots. However, such a method is limited since it requires the imitated motions to be slow [14], is limited to imitation and has no further creation, and is only applied on legged robots which are utilizing ZMP to stabilize. Cassie, with line feet, cannot use such a method [16].

To address Prob. 3, feedback control approaches on Cassie are introduced. Cassie is a bipedal robot manufactured by Agility Robotics. A framework for nonlinear optimization for Cassie based on hybrid zero dynamics (HZD) [17] is constructed and solved in [18]. Standing and walking controllers for Cassie using a full-model gait library are developed in [19]. A reduced-model-based walking controllers are designed in [20], [21]. However, all of the previous controllers for Cassie are unable to change its height above the ground while walking, and more importantly, there has been no effort to design emotive behavior design for Cassie.

B. Contributions

This paper makes the following contributions:

- A dynamic relatable robotic character using an animatronic bipedal robot Cassie is firstly introduced.
- A virtual character which agrees with holonomic constraints for Cassie is created in Blender, a 3D creation suite.
- An original nonlinear constrained optimization schematic is realized to translate the animated kinematic motions to be dynamically feasible for legged robots while staying close to the animated motion.
- A versatile walking controller which is able to change walking height, besides the features developed in [19],

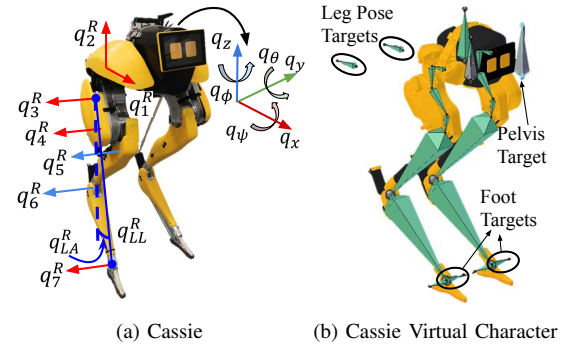


Fig. 3: (a) Cassie with eye-tablet in real world with the generalized floating-based coordinates and the definition of robot virtual leg length $q_{LL}^{L/R}$ and the leg angle $q_{LA}^{L/R}$. (b) Modelling and rigging of Cassie virtual character in Blender.

is developed to enable Cassie to perform agile and complex designed motions.

- An end-to-end framework, as shown in Fig. 2, is presented to enable animatronic dynamic legged robots to have personalities and emotive expressions. Experimental validation on Cassie is presented.

C. Paper Structure

The framework developed in this paper is illustrated in Fig. 2. We first introduce animation techniques to design the behaviors for the Cassie virtual character to create expressive and emotive robot motions. This solves Prob. 1 and is presented in Sec. II. However, the resulting motions may not be dynamically feasible for the robot. Therefore, Sec. III addresses Prob. 2 to translate the animation trajectories from the virtual character to the physical robot with dynamics constraints. Later, a motion library containing multiple story arcs is stored. To address Prob. 3, a controller and an automaton for dynamic walking robot characters are later developed in Sec. IV, to enable Cassie to perform agile motions from the generated library emotively. Experiments are conducted in Sec. V to validate proposed methodology and conclusions and future work is discussed in Sec. VI.

II. ANIMATION CREATION

Having provided an introduction and overview on our work, we now present details on how to create a Cassie virtual character and design emotive behaviors in animations for the virtual character.

A. Physical Robot to Virtual Character

Animation creation bridges animators who may have little robotics background with the robots. Creating a virtual character for Cassie is the first step to utilize these animation techniques. In this paper, we are using Blender[®] which is an open-source 3D creation suite and provides tools for modelling, rigging, animation, rendering, etc.

In real life, Cassie, as shown in Fig. 3a, has 10 actuated rotational joints $q_{1,2,3,4,7}^{L/R}$ and 4 passive joints $q_{5,6}^{L/R}$ on its Left/Right legs. Counting the 3 transition DoF (sagittal, lateral, horizontal) $q_{x,y,z}$ and 3 rotation DoF (roll, pitch, yaw)

q_ψ, θ, ϕ on its pelvis, it has 20 DoF in total. To be concise, we denote the generalized coordinates of Cassie as:

$$\mathbf{q} = [\mathbf{q}^p, \mathbf{q}^m, q_5^L, q_6^L, q_5^R, q_6^R]^T$$

where

$$\begin{aligned} \mathbf{q}^p &= [q_x, q_y, q_z, q_\psi, q_\theta, q_\phi]^T, \\ \mathbf{q}^m &= [q_1^L, q_2^L, q_3^L, q_4^L, q_7^L, q_1^R, q_2^R, q_3^R, q_4^R, q_7^R]^T, \end{aligned}$$

and $\mathbf{q}^p, \mathbf{q}^m$ denote pelvis DoFs and actuated motors positions, respectively.

We also denote $\mathbf{q}_i = [q_i^L, q_i^R]^T$ as the i -th joints on both legs, and $\mathbf{q}_{LL} = [q_{LL}^L(q_3^L, q_4^L), q_{LL}^R(q_3^R, q_4^R)]^T$ and $\mathbf{q}_{LA} = [q_{LA}^L(q_3^L, q_4^L), q_{LA}^R(q_3^R, q_4^R)]^T$ as virtual leg length and leg pitch, respectively. Moreover, we define $\mathbf{x}_{foot}^{L/R}(\mathbf{q}) = [x_{foot}^{L/R}, y_{foot}^{L/R}, z_{foot}^{L/R}]^T$, the Cartesian positions of robot's end-effectors which are the feet on the Left/Right legs. We denote $\mathbf{x}_{CoM}(\mathbf{q}) = [x_{CoM}, y_{CoM}, z_{CoM}]^T$ as Cassie's Center-of-Mass (CoM) Cartesian position. These terms can be determined by forward kinematics.

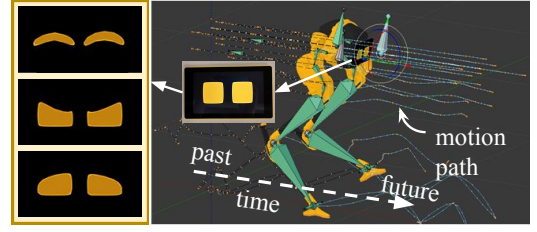
Based on this, we created a rigged character model of Cassie in the virtual world, as illustrated in Fig. 3b. Rigging involves taking the geometric information from each link on Cassie and mapping the rigid body as a "bone" (green link in Fig. 3b) and kinematic connection as "joint" which connects two adjacent bones. While animators can design a motion by specifying sequences of joint positions, it is more efficient and natural for them to specify the poses of end effectors (either the feet on the ground or the pelvis). Towards this, we added target bones such as a *Pelvis Target (PT)* which determines the pelvis pose, *Leg Pose Targets (LPTs)* to change leg's orientation, and *Foot Targets (FTs)* to position two feet, as presented in Fig. 3b.

B. Inverse Kinematics Constraints

Once animators have defined the pose of the target bones, like placed Foot Targets (FTs) on the ground and tilted Pelvis Target (PT) to make Cassie look in some direction, the rest of joint positions can be determined by solving an inverse kinematics (IK) problem. However, as mentioned, two joints corresponding to the joints from knee to shin \mathbf{q}_5 and the ankle joint \mathbf{q}_6 are passive and are connected through stiff leaf springs. When the springs are uncompressed, there exists additional constraints related to the knee and ankle joints, specifically, $\mathbf{q}_5 \equiv 0$, $\mathbf{q}_6 + \mathbf{q}_4 - 13^\circ = 0$ [19].

Therefore, the inverse kinematics problem can be formulated as:

$$\begin{aligned} \min_{\mathbf{q}} \quad & \|\mathbf{x}_{foot}^{L/R}(\mathbf{q}) - \mathbf{x}(FTs)\|_2^2 \\ \text{s.t.} \quad & \mathbf{q}_{min} \leq \mathbf{q} \leq \mathbf{q}_{max}, \\ & \mathbf{q}^p - p(PT) = 0, \\ & \mathbf{q}_2 - f(LPTs) = 0, \\ & \mathbf{q}_5 = 0, \\ & \mathbf{q}_6 + \mathbf{q}_4 - 13^\circ = 0, \end{aligned} \quad (1)$$



(a) Eyes Expressions and Animation Sketching



(b) Rendered Happy Motion

Fig. 4: Animation of the Cassie virtual character. (a) Walking animation where the motion path presents the trajectories of each joint of the character along time. On the left hand side shows examples of visual eye expressions (from top to bottom: happy, angry, sad). (b) The designed motion, such as happy motion, can be later rendered and produce an animation.

where the objective function is the L_2 distance between the Cartesian positions of the robot's feet $\mathbf{x}_{foot}^{L/R}$ and Foot Targets $\mathbf{x}(FTs)$. Pelvis pose \mathbf{q}^p is constrained to follow the pose of Pelvis Target $p(PT)$. $f(LPTs)$ captures the relative rotation between the Leg Pose Target (LPT) to its corresponding leg orientation in that leg's horizontal planes. The constraint of $\mathbf{q}_2 - f(LPTs) = 0$ enforces each leg to always point towards its LPT.

As a result, joint trajectories which satisfy Cassie's kinematic constraints are obtained by solving optimization problem (1) during the animation creation. We develop a custom solver as a Python plugin for Blender as its IK solver does not support internal joint constraints.

C. Creating Animation

To provide visual cues of emotion for the character, we also added eyes for the virtual character as illustrated in Fig. 4a. The eye motion can be designed jointly with the body motion. In real life, we added a tablet to display visual eye expressions as shown in Fig. 3a. Once we have a satisfying virtual character, animators could be given free rein to design emotive and expressive behaviors for the character. Animation technique allows the animator to focus on the creation of motion through creation of *keyframes* with the poses in the frames between two adjacent keyframes interpolated automatically. The animator is free to choose different methods of interpolation, such as Bézier or linear, provided by Blender. A typical walking animated motion is shown in Fig. 4a. Please note that the keyframe is more heavily weighted than the interpolated frame in terms of motion design in the animation field. This property will be used in the following optimization in Sec. III.

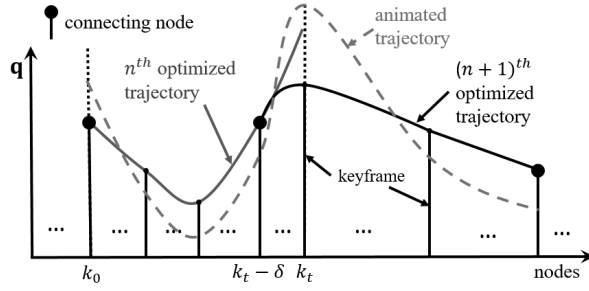


Fig. 5: Schematic for animation-to-dynamics optimization in the standing scenario. During optimization, the arbitrarily-long animated trajectory is broken into sub-motions and optimization is applied to each smaller trajectory in sequence. Between each sub-motion, such as the one from k_0 -th node to $(k_t - \delta)$ -th node, smoothness constraints are applied on the connecting node.

Once an animated motion is completed, the animator is able to utilize the render engine in Blender to create a professional animated movie, like Fig. 4b. Moreover, we provide a tool for animators to specify the mode, *i.e.* standing or walking, of the designed motion, and to extract and store all joint trajectories in the animation they created. We denote the animation as \mathcal{Q}_{ani} with sequence of frames $\{\mathbf{q}_{ani}\}$.

III. ANIMATION-TO-DYNAMICS OPTIMIZATION

Having created a pipeline to create animation behaviors for Cassie, we will next see how to transfer these to hardware. While we included an IK optimization in Sec. II-B to ensure the animated motion is kinematically feasible for Cassie, however, the created motion may still may violate certain laws of physics. Therefore, the objective of this section is to optimize for a trajectory that is dynamically-feasible, enforced by various constraints, while also being close to the animation.

We propose to solve this by formulating a direct-collocation based nonlinear optimization. However, the character's motion could either be in double-support standing or could be walking, wherein the dynamics constraints are different. Moreover, the input animated motion can be arbitrarily long, *i.e.*, could even be tens of seconds (resulting in thousands of collocation nodes), and direct-collocation at that large time spans is computationally hard to optimize. Therefore, we address this by firstly separating the optimization problem into standing and walking scenarios where distinct optimization solutions are discussed, and secondly by splitting the optimization into shorter time spans with continuity constants between each.

A. Optimization for Standing

The input animated motion \mathcal{Q}_{ani} is a sequence of nodes $\{\mathbf{q}_{ani}\}$ wherein each node is the full state of the animated Cassie character with a timestamp. As the input animation has arbitrarily long frames, the sequence \mathcal{Q}_{ani} elapsed in seconds could contain thousands of nodes. In order to make the optimization problem solvable, we break the optimization into multiple smaller ones between each sub-motion while

Algorithm 1 Optimization for Standing Motion

Input: Animated Motion $\mathcal{Q}_{ani} = \{\mathbf{q}_{ani}(k)\}$, $k = 1, 2, \dots, N$
Output: Optimized Motion $\mathcal{Q}_{opt} = \{\mathbf{q}_{opt}(k)\}$, $k = 1, 2, \dots, N$
Initialization: $(k_0, k_t) \leftarrow (1, 1 + n)$, $\mathcal{Q}_{opt} = \emptyset$
1: **while** $k_t \leq N$ **do**
2: $\{\mathbf{q}_{opt}\} \leftarrow$ solution of:
 $\min_{\{\mathbf{q}\}} \sum_{i=k_0}^{k_t} \|\mathbf{u}(i)\|_2^2 + c \|\mathbf{q}_{ani}(i) - \mathbf{q}(i)\|_2^2$ (2)
s.t. $t_{ani} \leq t_{opt} \leq 4t_{ani}$ (2.1)
 $\mathbf{q}_{min} \leq \mathbf{q} \leq \mathbf{q}_{max}$ (2.2)
 $\mathbf{u}_{min} \leq \mathbf{u} \leq \mathbf{u}_{max}$ (2.3)
 $D(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + G(\mathbf{q}) = B\mathbf{u} + \lambda J^T(\mathbf{q})$ (2.4)
 $J(\mathbf{q})\ddot{\mathbf{q}} + \dot{J}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = 0$ (2.5)
 $\mathbf{x}_{foot}^R(\mathbf{q}) = [0, 0, 0]^T$ (2.6)
 $[-10, -10, 0]^T \leq \mathbf{x}_{foot}^L(\mathbf{q}) \leq [10, 10, 0]^T$ (2.7)
 $F_c(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ stays in the friction cone ($\mu = 0.6$) (2.8)
 $\mathbf{x}_{CoM}(\mathbf{q})$ lies in the support region (2.9)
 $\|\mathbf{q}_{ani}^p(k_0) - \mathbf{q}^p(k_0)\|_2^2 = 0$ (2.10)
 $\|\dot{\mathbf{q}}_{ani}^p(k_0) - \dot{\mathbf{q}}^p(k_0)\|_2^2 = 0$ (2.11)
 $\|\dot{\mathbf{q}}(k_t)_{x,y,z}\|_2 = 0$ if $k_t = N$ (2.12)
3: $(k_0, k_t) \leftarrow (k_t - \delta, k_t - \delta + n)$
4: $\mathcal{Q}_{opt} = \mathcal{Q}_{opt} \cup \{\mathbf{q}_{opt}\}$
5: **end while**
6: **return** \mathcal{Q}_{opt}

maintaining the smoothness of the entire motion, as illustrated in Fig. 5 and explained in Algorithm 1.

In Algorithm 1, we cut the long sequence of nodes whose horizon is N into sub-motions which have n nodes for each. As the keyframe is more important than the interpolated frame in the animation, we want the start and end nodes to be keyframes, and to have similar numbers of keyframes among sub-motions. Therefore, the value of n is different in each sub-motion. On each sub-motion, the optimization problem formulated as (2) in Algorithm 1 is solved. The objective function (2) is set to minimize the sum of energy consumption and the L_2 distance between the optimized poses \mathbf{q} to the animated ones \mathbf{q}_{ani} in the sub-horizon. The parameter c has three major roles: to weigh between the similarity to the animated motion and total energy consumption, to emphasize the poses of the animation keyframe in the optimized motion, and to push the end of the sub-motion to get as close as possible to the animated one. Therefore, c is set to 1500, 10^4 , 10^5 at normal frames, keyframes, and the ending frame, respectively.

Besides putting boundaries on the elapsed time (2.1), states (2.2), and inputs (2.3), (2.4) enforces Cassie's dynamics while (2.5-2.7) formulate the kinematics constraints and place the left and right foot Cartesian positions $\mathbf{x}_{foot}^{L/R}$ on the ground. Moreover, (2.8) prevents the feet slipping on the ground whose static friction ratio μ is 0.6, and (2.9) balances the robot by limiting the CoM position \mathbf{x}_{CoM} projected on the ground lying in the support region composited by two feet $\mathbf{x}_{foot}^{L/R}$ as formulated in [18].

Furthermore, in order to connect two adjacent sub-motions, we added (2.10) to enforce the start node of current sub-motion to overlap with the end node of the last subsequent motion, while adding velocity constraints (2.11) to smooth the connection, as illustrated in Fig. 5.

Please note that we only restrict the smoothness of the pelvis motion \mathbf{q}^p and let the kinematics constraints solve the rest of joints. Also, to leverage the smoothness constraints, we begin from δ number of frames ahead of the end node of the previous sub-motion when sliding through the entire motion. In practice, δ varies to ensure each sub-motion starts with a keyframe. Note that (2.12) is also added for zero-transition-velocity of the pelvis at the ending of the entire motion.

Therefore, in this way, we can obtain optimized motions with arbitrary time spans as the algorithm only focuses on a short but optimize-able long motion and the connection points in between.

B. Optimization for Walking

For a walking robot, a gait g is defined as periodic joint position and velocity profiles for all actuated joints. For each gait, there is a gait parameter p that identifies it. It is hard to require Cassie to mimic whole body motion while maintaining stable walking gaits using a real-time controller. Therefore, we mediate the motion imitation and nonlinear optimization by limiting the imitated DoF to the gait parameter which consists of sagittal and lateral walking velocities, v_x and v_y , and walking height h (the height of pelvis above ground) from the animation, *i.e.*, $p = [v_x, v_y, h]^T$. In this way, we choose to design a library of periodic walking gaits for different gait parameters, and let Cassie track the animated ones.

The optimization is applied on a hybrid model which contains two continuous domains, *right stance* and *left stance*, and two transition domains called *left impact* and *right impact*. During each stance domain, one of the legs is pinned on the ground. Domain transitions are triggered at foot impact events causing the model to transit from *right stance* to *left stance* and *left stance* to *right stance*, respectively. Periodic walking motion can be realized by starting and ending in *right stance*.

The optimization formulation to generate walking gaits for Cassie is described in [19]. It includes constraints on average velocity to match the given gait parameter, virtual constraints to have hybrid zero dynamics which is impact-invariant [17], and others. For solving for gaits with a desired walking height h for Cassie, we include two more constraints:

- Pelvis height: $q_z \in [h - 0.035, h + 0.035]$ m
- Step time: $t \in [t_{min}(h), 0.4]$, where $t_{min}(h)$ is linearly interpolated from 0.2s to 0.4s with respect to h

In this way, we generated a gait library \mathcal{G} for the following gait parameters p :

- 11 forward/backward walking speeds v_x in the range $[-1, 1]$ m/s
- 11 left/right side-stepping speeds v_y in the range $[-0.3, 0.3]$ m/s
- 11 walking heights h in the range $[0.65, 1]$ m per step

This gait library has $11 \times 11 \times 11 = 1331$ gaits parametrized by Bézier polynomials and consists of a set $\mathcal{G} = \{g_i(t_i, \alpha_i^j)\}$ where α_i^j represents the coefficients of the Bézier polynomial for the j -th motor in the i -th gait g_i and t_i encodes the period for the gait.

C. Motion Library

Based on the animation creation process, we created multiple animations with different emotions (happy, sad, tired, etc.). For each animated motion, the above-proposed optimizations are applied to obtain dynamically feasible motion for Cassie that is close to the animated motion. Also, a tag indicating the motion's emotion, as well as the control mode, *i.e.* standing or walking, is added. In this way, we store the optimized motions in a motion library from which they can be retrieved by their unique IDs.

IV. CONTROL AND AUTOMATON FOR ROBOT CHARACTER

Once a library of dynamically feasible motions is obtained, we need a real-time controller for Cassie to replay the input motions. Moreover, switching between different controllers, *e.g.* from standing to walking, or performing different motions in sequence, could destabilize the robot if the transition happens inappropriately. Therefore, a high-level automaton is needed to decide which motion to send to the controller, and when and how the transition should start and end.

A. Versatile Standing and Walking Controller for Cassie

Basic standing and walking controllers for Cassie are developed in [19]. However, the walking controller is not able to reliably track the velocity commands in the lateral direction and at different walking heights. Therefore, based on the previous work, we developed a standing controller, as shown in Fig. 6a which is capable of tracking all 6 DoF motions for the pelvis, as well as a controller that is able to significantly change body height while walking in 3D space, as illustrated in Fig. 6b.

1) *Standing Controller*: The reference \mathbf{q}_d^m and $\dot{\mathbf{q}}_d^m$ is the motions of all the motors from the motion library. However, the foot joint positions do not track the animated motion but rather help in keeping the feet on the ground and in rejecting perturbation to keep the sagittal component of the Center-of-Mass \mathbf{x}_{CoM} to stay in the support region, *i.e.*, $\delta x_{CoM} = \|\mathbf{x}_{CoM}(1) - \frac{1}{2}(\mathbf{x}_{foot}^R(1) + \mathbf{x}_{foot}^L(1))\|_2 \equiv 0$. For every actuated joint, a PD controller is applied to track the reference joint trajectories based on the feedback $\hat{\mathbf{q}}^m$, $\dot{\hat{\mathbf{q}}}^m$ from the encoders.

2) *Walking Controller*: A desired gait g_d is obtained by linearly interpolating the gait library \mathcal{G} obtained in Sec. III-B with respect to the observed current robot's gait parameters $\hat{p} = [\hat{v}_x, \hat{v}_y, \hat{h}]^T$. To maintain a stable walking gait and track the reference gait parameter p_d , a gait regulator is applied to the desired gait. The gait regulator has a correction δg_d with three terms defined as:

$$\delta g_d = \begin{bmatrix} \delta q_3^{sw} \\ \delta q_1^{sw} \\ \delta q_4^{st} \end{bmatrix} = K_P(h^d)(p^d - p) + K_D(h^d)(\dot{p}_d - \dot{p}') \quad (3)$$

where δq_{LA}^{sw} , δq_1^{sw} are corrections for the leg pitch and abduction joint of the *swing* leg, and δq_{LL}^{st} represents the correction to the *stance* leg length. Also, p^d and \hat{p} are the desired and current gait parameters, while \dot{p}' is the numerical

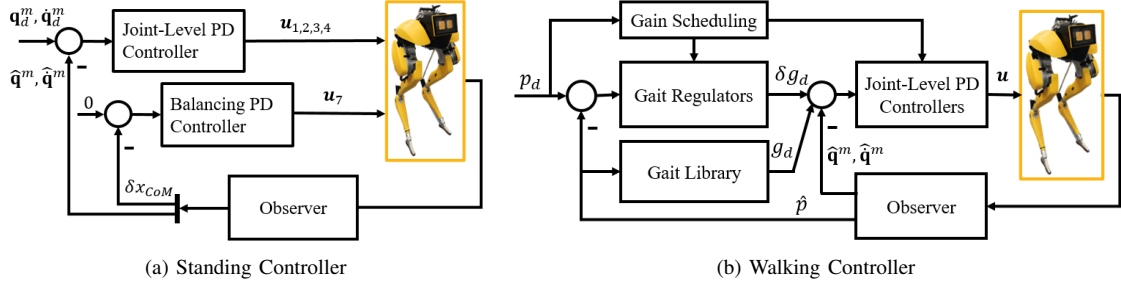


Fig. 6: Control structure for animated Cassie. (a) The input to the standing controller is the joints trajectories from the motion library. (b) The walking controller is designed to track input gait parameter p_d which is the sagittal and lateral velocity and desired walking height. It interpolates the gait library by observed gait parameter \hat{p} to obtain a reference gait g_d and regulate this by a gait regulator. The input walking height is also fed forward to a gain scheduler which changes the parameters used in the controllers.

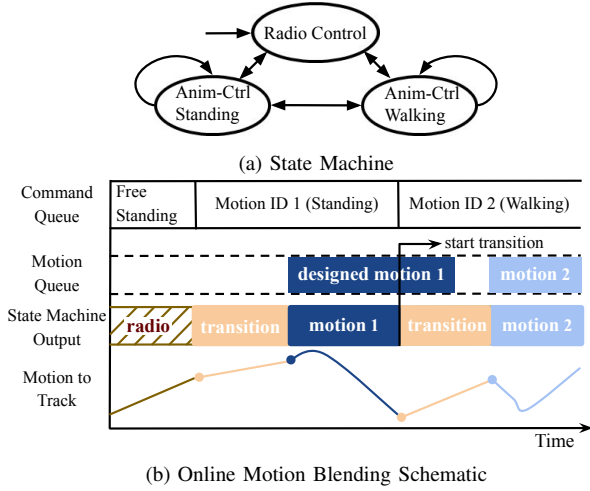


Fig. 7: Automaton for robot character. (a) A state machine is running in real time to concatenate and smooth the reference motions. (b) The output from the state machine is a blended motion with motions from motion library or radio commands and transitions in between. Such motion is sent to controller to track.

differentiation of \hat{p} . Also, we map the walking height h to the stance leg length q_{LL}^{st} with a constant offset.

In this way, the commanded gait becomes $g_d + \delta g_d$ which is the sum of the reference gait g_d and regulating term δg_d , with proper dimension. Later, joint-level PD controllers applies input torque commands to each motor, based on the measured joint states $\hat{q}^m, \dot{\hat{q}}^m$.

This cascade walking controller has a gain scheduling module wherein the PD gains are scheduled based on the reference walking height. This is achieved by linearly interpolating in a look-up parameter table with respect to desired walking height h_d .

B. Automaton for Robot Character

When we splice multiple motions together, there is no guarantee that the connecting point is continuous when we switch from one motion to the next. Such discontinuity may cause the robot to fall. Also, to ensure safety, radio commands are needed for human operators to take over the control of the robot. Switching between different animated motions, as well as changing between animation-controlled

mode and manual radio-controlled mode, may drive the robot to fall and thus such discontinuity should be smoothed in real time. To address this problem, we propose an online automaton, as shown in Fig. 7a, that is capable of concatenating two motions and producing a smooth transition accounting for robot's current state.

Note that the radio can command Cassie to stand or walk, and each motion comes with a control mode. Therefore, the transition types between adjacent states are within the Cartesian product of the modes of $\{Standing, Walking\}$ and $\{Animation-Controlled, Radio-Controlled\}$. The four elemental types of transitions for the animatronic walking automaton and the transitions can be defined as:

- *Stand to Walk*: Change the standing height to the desired walking height, then lean Cassie's pelvis to left and lift its right leg to start walking.
- *Walk to Stand*: Decelerate robot's velocity to zero and switch to standing mode when a step is completed.
- *Radio from/to Animation*: Transit from current states to the desired states which are the first frame in animation or commands from the radio. The state trajectories are obtained by linear interpolation.
- *Animation to Animation*: This transition happens when the motion ID is changed. States are commanded to travel by linearly interpolating current state and the first frame of the next motion.

As a result, the transitions in the state machine shown in Fig. 7a can be determined by combining the elemental transitions pairwise.

As suggested in Fig. 2 and Fig. 7b, the real-time automaton for the animatronic walking robot works as follows: 1) a motion ID, determined by a queue of motion IDs, or by requests from the interacting human, is sent to the state machine in Fig. 7a; 2) the state machine decides whether it needs a transition, what kind of transition should take place, and when should the designed motion start, based on the real-time feedback from the robot, radio commands, and input motion ID; and 3) the motion to track is then composed by the transitions and interpolating the designed motion and is sent to the real-time controller for Cassie to track.

In this way, an end-to-end framework for the animatronic walking robot Cassie is built up and the requirements for

Cassie’s dynamically feasible expressive motions are met.

V. RESULTS AND EXPERIMENTS

A. Optimization Results

We formulated the optimization problems in Sec. III by using the open-sourced optimization toolkit CFROST [18] and solved by IPOPT [22]. The proposed standing optimization scheme shows the capacity to obtain dynamically feasible motion while minimizing the deviation from the animated trajectories for animations with any duration. A 5-second standing motion can be solved within an hour offline, using the full-order model of the robot. As a test case we compare the animated motion and optimized motion on the same standing controller introduced in Sec. IV-A in a high-fidelity physical simulator of Cassie built in Simulink. The result is demonstrated in Fig. 8. It shows that the animated motion, shown as a dash line in Fig. 8a,8b causes Cassie to fall as illustrated in Fig. 8c. This is because the animated motion changes so fast, *i.e.* pelvis yaw q_ϕ changes over 50 degrees within 1 second at 13s, which is beyond what the robot can achieve. In contrast, in Fig. 8d, Cassie successfully performs the entire motion after using the optimized motion which is the solid line in Fig. 8a,8b. Although the elapsed time of the optimized motion is enlarged 2 times compared to the animated one, such change of time span smooths the motion while staying close to the animated motion profile. Such motion profile encodes expressive contents in the animation while also being dynamically feasible. Further, we notice that although there is no motion in the pelvis roll in the animation, the optimization adds a significant movement on that DoF to compensate the fast change of motions from other DoFs. A similar change happens in the lateral transition, though the magnitude is small. This is the reason the optimized motion will not cause failure even though the motion is as aggressive as the animated one. Also note that the entire motion to optimize is 15 seconds and contains 373 nodes. If the animated motion is directly input to the optimization solver without our proposed method, the solver wouldn’t be able to find a valid solution.

B. Experimental Validation

We deployed and tested the entire proposed pipeline on Cassie. The automaton and gait library interpolation is running at 200Hz and the rest of standing and walking controllers is in a 2kHz loop. We created a story where Cassie shows curiosity by chasing a laser point moving on the ground. This story includes three pieces of aggressive sub-motions, standing-walking-standing, to push the limit of Cassie’s mobility. We also created a library containing motions with various emotional attributes, such as happy, sad and tired. We used the laser-chasing story as an illustrative example of the standing-walking-standing motion transition. In this section, we discuss this experiment. The experimental video¹ includes the entire story and other emotive motions on Cassie.

¹<https://youtu.be/zjzfzfgb3oK8>

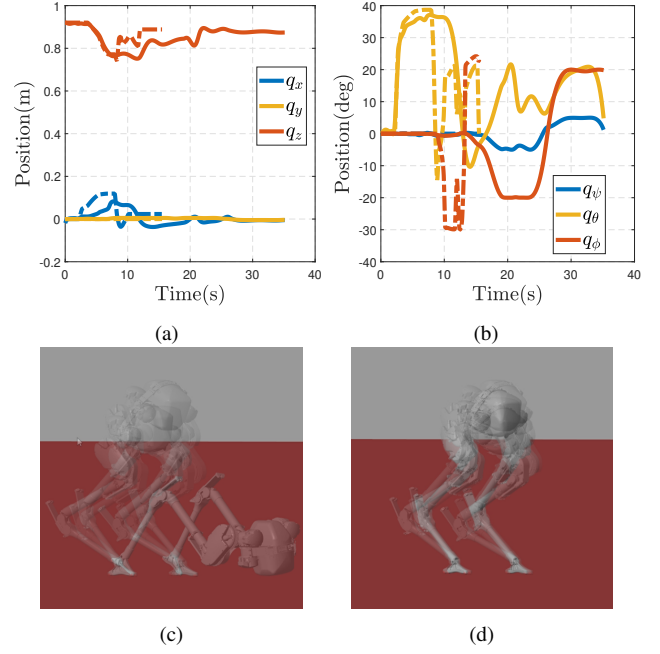


Fig. 8: Results of the optimization for standing. (a)(b) Comparisons of resulted pelvis transitional and rotational position \mathbf{q}^p profiles from animation and optimization. The motion from animation is drawn in dash line while its optimized motion is presented in solid line. (c) Dynamic simulation of Cassie with the animated motion results in a fall. (d) No failure happens after the animated motion is optimized using the same controller and dynamic simulator.

Standing Section 1: As shown in Fig. 9a, in the first standing sequence, Cassie sees a laser point on the ground and leans, turns, and squats to keep it in sight. Such animated movement is aggressive but was optimized to be dynamically feasible.

Walking Section: Cassie then transitions into a walking motion, walking forward while squatting lower while walking to chase the laser. No previous work has been able to adjust Cassie’s walking height, as shown in Fig. 9b.

Standing Section 2: Finally, Cassie decelerates and stops walking autonomously, transitions back to standing and performs a final standing motion where it looks around, as illustrated in Fig. 9c.

According to the experiments, people can clearly infer the robot’s emotions and the entire story idea as the motions are designed to be emotive and eye motions are animated and replayed jointly. Moreover, without the proposed automaton, the transitions between standing and walking could cause failure on the robot as there exists discontinuities in robot states and control policies during mode switching. Therefore, although the original animated motion is arbitrarily aggressive, Cassie is able to mimic its virtual character’s motion while maintaining balance and retaining the emotive content from the animation using our proposed pipeline.

VI. CONCLUSIONS AND FUTURE WORK

To our knowledge, this paper is one of the first to present a dynamic relatable robotic character to behave with agility,

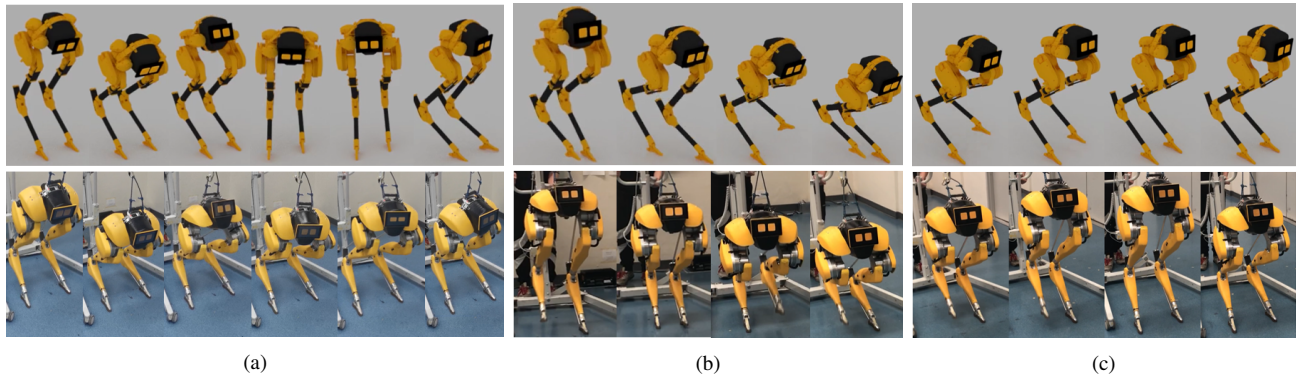


Fig. 9: Snapshots of Cassie performing curiosity (laser chasing) story. The top half is the original animation and the bottom half is the motions performed by Cassie in real life. (a) Cassie looks on the ground when it is standing. (b) Cassie starts to walk to chase the laser while lowering its walking height. (c) Cassie decelerates and transitions to standing and looks around. The designed eyes videos are also played synchronously with the body motions.

emotions and personalities based on animated motions of a virtual character. In this work, a virtual character that enforces the kinematic constraints of Cassie is created to bridge the animation technology and robotics. A segmented optimization framework is developed to translate arbitrarily long animated motions to be dynamically feasible for the legged robot in both standing and walking scenarios. A versatile walking controller which is able to change the walking height for Cassie is also realized. An end-to-end pipeline centered on a state machine is developed to concatenate different motions from a motion library based on the input commands and feedback from the robot in real time. Experiments are conducted and the animated Cassie shows a capacity to perform motions with emotional attributes.

In the future, the motion library needs to include more animated and spirited motions enabling human-dynamic-robot interaction based on the emotive robot behaviors created by our proposed animation approach.

REFERENCES

- [1] G. Venture and D. Kulić, "Robot expressive motions: a survey of generation and evaluation methods," *ACM Transactions on Human-Robot Interaction*, vol. 8, no. 4, pp. 1–17, 2019.
- [2] (2020) Meet cozmo. [Online]. Available: <https://anki.com/en-us/cozmo.html>
- [3] A. K. Pandey and R. Gelin, "A mass-produced sociable humanoid robot: pepper: the first machine of its kind," *IEEE Robotics & Automation Magazine*, vol. 25, no. 3, pp. 40–48, 2018.
- [4] H. Ishiguro, T. Ono, M. Imai, T. Maeda, T. Kanda, and R. Nakatsu, "Robovie: an interactive humanoid robot," *Industrial robot: An international journal*, 2001.
- [5] C. D. Kidd, W. Taggart, and S. Turkle, "A sociable robot to encourage social interaction among the elderly," in *IEEE International Conference on Robotics and Automation*, Orlando, USA, 2006.
- [6] J. Kedzierski, R. Muszynski, C. Zoll, A. Oleksy, and M. Frontkiewicz, "Emys—emotive head of a social robot," *International Journal of Social Robotics*, vol. 5, no. 2, pp. 237–249, 2013.
- [7] J.-C. Latombe, *Robot motion planning*. Springer Science & Business Media, 2012, vol. 124.
- [8] A. van Breemen, "Animation engine for believable interactive user interface robots," in *IEEE RJS International Conference on Intelligent Robotics and Systems*, Sendai, Japan, 2004, pp. 2873–2878.
- [9] (2020) andyrobot. [Online]. Available: <https://http://andyrobot.com/>
- [10] S. Hoshyari, H. Xu, E. Knoop, S. Coros, and M. Bächer, "Vibration-minimizing motion retargeting for robotic characters," *ACM Transactions on Graphics*, vol. 38, no. 4, pp. 1–14, 2019.
- [11] J. van Breemen and Y. Xue, "Advanced animation engine for user interface robots," in *IEEE RJS International Conference on Intelligent Robotics and Systems*, Beijing, China, 2006, pp. 1824–1830.
- [12] L. Takayama, D. Dooley, and W. Ju, "Expressing thought: improving robot readability with animation principles," in *Proceedings of the 6th international conference on Human-robot interaction*, 2011.
- [13] W. Suleiman, E. Yoshida, F. Kanehiro, J.-P. Laumond, and A. Monin, "On human motion imitation by humanoid robot," in *IEEE International Conference on Robotics and Automation*, 2008, pp. 2697–2704.
- [14] J. Koenemann, F. Burget, and M. Bennewitz, "Real-time imitation of human whole-body motions by humanoids," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 2806–2812.
- [15] Y. Ishiguro, K. Kojima, F. Sugai, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Bipedal oriented whole body master-slave system for dynamic secured locomotion with lip safety constraints," in *IEEE/RIS International Conference on Intelligent Robots and Systems*, 2017.
- [16] M. Vukobratovic and B. Borovac, "Zero-moment point—thirty five years of its life," *International journal of humanoid robotics*, vol. 1, no. 01, pp. 157–173, 2004.
- [17] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2018.
- [18] A. Hereid, O. Harib, R. Hartley, Y. Gong, and J. W. Grizzle, "Rapid bipedal gait design using c-frost with illustration on a cassie-series robot," *arXiv preprint arXiv:1807.06614*, 2018.
- [19] Y. Gong, R. Hartley, X. Da, A. Hereid, O. Harib, J.-K. Huang, and J. Grizzle, "Feedback control of a cassie bipedal robot: Walking, standing, and riding a segway," in *American Control Conference*, 2019.
- [20] X. Xiong and A. D. Ames, "Coupling reduced order models via feedback control for 3d underactuated bipedal robotic walking," in *IEEE-RAS International Conference on Humanoid Robots*, 2018.
- [21] T. Apgar, P. Clary, K. Green, A. Fern, and J. W. Hurst, "Fast online trajectory optimization for the bipedal robot cassie," in *Robotics: Science and Systems*, vol. 101, 2018.
- [22] L. T. Biegler and V. M. Zavala, "Large-scale nonlinear programming using ipopt: An integrating framework for enterprise-wide dynamic optimization," *Computers & Chemical Engineering*, vol. 33, no. 3, pp. 575–582, 2009.