Supplemental Document for

Computational Design of Walking Automata

Gaurav Bharaj[1]* Stelian Coros[2] Bernhard Thomaszewski[3] James Tompkin[1] Bernd Bickel[4] Hanspeter Pfister[1]
[1]Harvard SEAS [2]Carnegie Mellon University [3]Disney Research Zürich [4]IST Austria

A Physical Simulation

Each automata is modeled as a rigid multi-body system. Since the mechanisms we optimize typically exhibit numerous kinematic loops, we opt for a maximal coordinates dynamics formulation. Therefore, the state of each rigid body i consists of position and orientation degrees of freedom $\mathbf{q}_i$, and their linear and angular velocity derivatives $\dot{\mathbf{q}}_i$. The vectors $\mathbf{q}$ and $\dot{\mathbf{q}}$ concatenate the states of all rigid bodies in the system.

We model joints, virtual motors, and frictional contacts using a set of constraints of the form $\mathbf{C}(\mathbf{q}) = \mathbf{0}$, and their time derivatives $\dot{\mathbf{C}}(\mathbf{q}) = \dot{\mathbf{C}}^d$ [Cline and Pai 2003]. According to the principle of *virtual work*, the constraints give rise to internal forces $\mathbf{f}_c = \mathbf{J}^T\lambda$, where $\mathbf{J}$ denotes the Jacobian $\frac{\partial C}{\partial \mathbf{q}}$, and λ are Lagrange multipliers that intuitively correspond to the magnitudes of the generalized forces needed to satisfy each constraint. To integrate the motion of the mechanisms forward in time, we must first compute the constraint forces $\mathbf{f}_c$. Without loss of generality, we can express their magnitudes implicitly as:

$$\lambda = -k_p\mathbf{C}(\mathbf{q}_{t+1}) - k_d(\dot{\mathbf{C}}(\mathbf{q}_{t+1}) - \dot{\mathbf{C}}^d) \tag{1}$$

where subscript t indicates the time instance, and the coefficients k_p and k_d allow us to set the relative stiffness of different types of constraints. A Taylor-series approximation of the position constraints allows us to express $\mathbf{C}(\mathbf{q}_{t+1})$ as:

$$C(\mathbf{q}_t + h\dot{\mathbf{q}}_{t+1}) \doteq \mathbf{C}(\mathbf{q}_t) + hJ^T\dot{\mathbf{q}}_{t+1} \tag{2}$$

where h denotes the time step. Using the chain rule, the time-derivative of the constraints can be written as $\dot{\mathbf{C}}(\mathbf{q}_{t+1}) = \mathbf{J}^T\dot{\mathbf{q}}_{t+1}$. This allows us to approximate Eq. 1 as:

$$\mathbf{J}\dot{\mathbf{q}}_{t+1} = -a\lambda - ak_p\mathbf{C}(\mathbf{q}_t) + k_d a\dot{\mathbf{C}}^d \tag{3}$$

where $a = \frac{1}{hk_p+k_d}$. Using the equations of motion of the multi-body system, the generalized velocities $\dot{\mathbf{q}}_{t+1}$ are given by:

$$\dot{\mathbf{q}}_{t+1} = \dot{\mathbf{q}}_t + h\mathbf{M}^{-1}(\mathbf{F}_{ext} + \mathbf{J}^T\lambda) \tag{4}$$

where $\mathbf{M}$ denotes the system's mass matrix, and the term $\mathbf{F}_{ext}$ stores the gravitational forces acting on the system. Multiplying Eq. 4 by $\mathbf{J}$, and combining the result with Eq. 3, results in the following system of equations that is linear in λ:

$$\mathbf{A}\lambda = \mathbf{b} \tag{5}$$

where $\mathbf{A} = h\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T + a\mathbf{I}$ and $\mathbf{b} = k_d a\dot{\mathbf{C}}^d - ak_p\mathbf{C}(\mathbf{q}_t) - \mathbf{J}\dot{\mathbf{q}}_t - h\mathbf{J}\mathbf{M}^{-1}\mathbf{F}_{ext}$. Because the constraint forces arising from frictional contacts are subject to inequality constraints, as discussed shortly, rather than solving Eq. 5 directly, we follow the work of Smith et al. [2012] and compute λ by solving a quadratic program:

$$\min_{\lambda} \frac{1}{2}(\mathbf{A}\lambda - \mathbf{b})^T(\mathbf{A}\lambda - \mathbf{b}) s.t. \mathbf{D}\lambda \geq \mathbf{0} \tag{6}$$

where the matrix $\mathbf{D}$ stores all the inequality constraints that need to be enforced. Once the constraint forces are computed, we use Eq. 4 to compute the generalized velocity term $\dot{\mathbf{q}}_{t+1}$, and the positional degrees of freedom $\mathbf{q}_{t+1}$ are integrated forward in time as described by Witkin [2001].

The derivation we provide here is related to methods implemented by some modern rigid body engines, such as the Open Dynamics Engine [Smith 2008]. However, rather than being restricted to working with ad-hoc parameters that hold little physical meaning, such as the Constraint Force Mixing term, Error Reduction Parameter and the Parameter Fudge Factor, we control the behavior of our simulations by manipulating the stiffness and damping parameters, k_p and k_d, which are set independently for each constraint type (as detailed below). In the limit, as k_p goes to infinity and k_d to 0 (i.e., infinitely stiff spring), this formulation remains well-defined, and corresponds to solving the constraints exactly. However, from the point of view of numerical stability, it is often better to treat the constraints as stiff implicit penalty terms.

Pin joints that allow a pair of components to rotate relative to each other about a pre-specified axis are implemented using two sets of constraints. First, we ensure that the coordinates of the pin coincide in world space using a vector-valued constraint of the form $\mathbf{C}(\mathbf{q}) = \mathbf{x}(\mathbf{q}_i(t), \mathbf{p}_i) - \mathbf{x}(\mathbf{q}_j(t), \mathbf{p}_j)$. Here, $\mathbf{x}(\mathbf{q}_a, \mathbf{p}) = \mathbf{t}_a + \mathbf{R}_a\mathbf{p}$ corresponds to the world coordinates of the point $\mathbf{p}$, $\mathbf{t}_a \in \mathbb{R}^3$ is defined as the position of center of mass of rigid body a, and $\mathbf{R}_a$ corresponds to its orientation. The location of the pin joint is defined by specifying the local coordinates of the pin, $\mathbf{p}_i$ and $\mathbf{p}_j$, in the coordinate frames of the two rigid bodies i and j that are connected to each other. To ensure that the two rigid bodies rotate relative to each other only about the pre-scribed axis, we use an additional vector-valued constraint, $\mathbf{C}(q) = \mathbf{R}_i\mathbf{n}_i - \mathbf{R}_j\mathbf{n}_j$, where $\mathbf{n}_i$ and $\mathbf{n}_j$ represent the coordinates of the rotation axis in the local coordinates of the two rigid bodies, and are set to $(0, 0, 1)^T$ for all our experiments. The k_p and k_d coefficients for the pin joint constraints are set to 10^8 and 10^4, respectively.

Motor constraints are used to mimic the effect of physical actuators. For this purpose, we prescribe the time-varying, desired relative angle between a select set of rigid body pairs. In particular, we assume that each limb of the mechanical toys has an input crank that operates relative to the main body. As we already employ pin joint constraints between these pairs of rigid bodies, the motor constraints directly measure the difference between their relative orientation and the target motor angle. The target motor angles are specified by *phase profile functions* $f(\alpha)$, as described by Coros et al. [2013]. The desired value for the time derivative of the constraint, $\dot{C}^d$, is set to $\dot{f}(\alpha)$, and it intuitively corresponds to the target velocity of the virtual motor. The k_p and k_d coefficients for the motor constraints are set to 10^8 and 10^5, respectively.

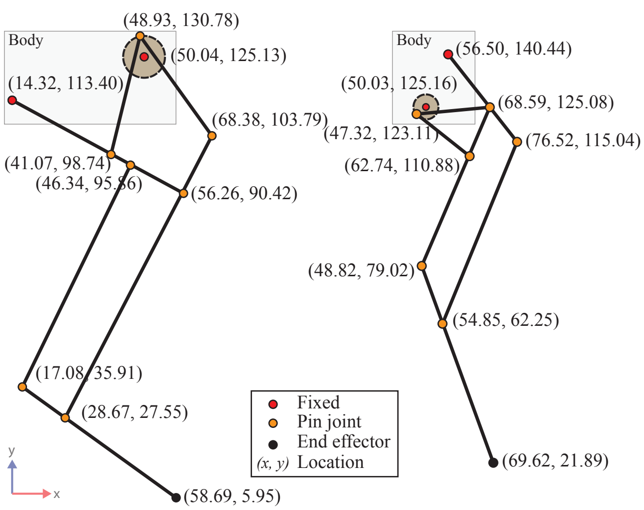
Frictional contacts move our automata around their simulated environments, and friction and contact forces must be bounded to generate physically-plausible results. Each contact introduces three constraints. Let $\mathbf{n}$ denote the contact normal. The first constraint specifies that the penetration distance, measured along the normal, should be 0: $C(\mathbf{q}_a) = \mathbf{n}^T(\mathbf{x}(\mathbf{q}_a, \mathbf{p}) - \mathbf{x}_p)$. Here, $\mathbf{p}_a$ corresponds to the coordinates of the contact point in the frame of rigid body a,

*Email: bharaj@g.harvard.edu

and $\mathbf{x}_p$ is the projection of the contact point onto the environment. For this constraint, $k_p = 10^8$, $k_d = 10^4$, and, importantly, the constraint force magnitude is constrained to be positive: $\lambda_n \geq 0$.

To model friction, we employ a pyramid approximation to the friction cone, as is standard in real-time simulation systems. More precisely, we let $\mathbf{t}_1$ and $\mathbf{t}_2$ be two orthogonal vectors that are tangent to the contact plane, and define constraints similar to the one for the normal direction, but acting along the tangent vectors. However, friction forces should only act to reduce the relative velocity at the contact point to 0. For this reason, we set k_p to 0 for these constraints, while k_d is set to 10^4. To ensure that tangential forces remain within the friction pyramid, we add inequality constraints of the form $-\mu\lambda_n \leq \lambda_t \leq \mu\lambda_n$ for the magnitude of the tangential forces acting along $\mathbf{t}_1$ and $\mathbf{t}_2$, where μ represents the friction coefficient.

B Linkage database



(a) Hind linkage (b) Front linkage

Figure 1: *Initial linkage configuration used for the* Dog *mechanical automata. Linkage (a) is also the linkage used for all legs (front and rear) for the* Giraffe-like *automata.*

References

CLINE, M., AND PAI, D. 2003. Post-stabilization for rigid body simulation with contact and constraints. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, vol. 3, 3744–3751 vol.3.

COROS, S., THOMASZEWSKI, B., NORIS, G., SUEDA, S., FORBERG, M., SUMNER, R. W., MATUSIK, W., AND BICKEL, B. 2013. Computational design of mechanical characters. *ACM Trans. Graph. 32*, 4, 83:1–83:12.

SMITH, B., KAUFMAN, D. M., VOUGA, E., TAMSTORF, R., AND GRINSPUN, E. 2012. Reflections on simultaneous impact. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012) 31*, 4, 106:1–106:12.

SMITH, R., 2008. Open dynamics engine. http://www.ode.org/.

WITKIN, A. 2001. Physically based modeling. Tech. rep., SIGGRAPH 2001. Course Notes.