

Designing Cable-Driven Actuation Networks for Kinematic Chains and Trees

Vittorio Megaro*
ETH Zurich, Disney Research

Espen Knoop*
Disney Research

Andrew Spielberg*
Massachusetts Institute of Technology

David I.W. Levin
University of Toronto

Wojciech Matusik
Massachusetts Institute of Technology

Markus Gross
ETH Zurich, Disney Research

Bernhard Thomaszewski
Disney Research

Moritz Bächer
Disney Research

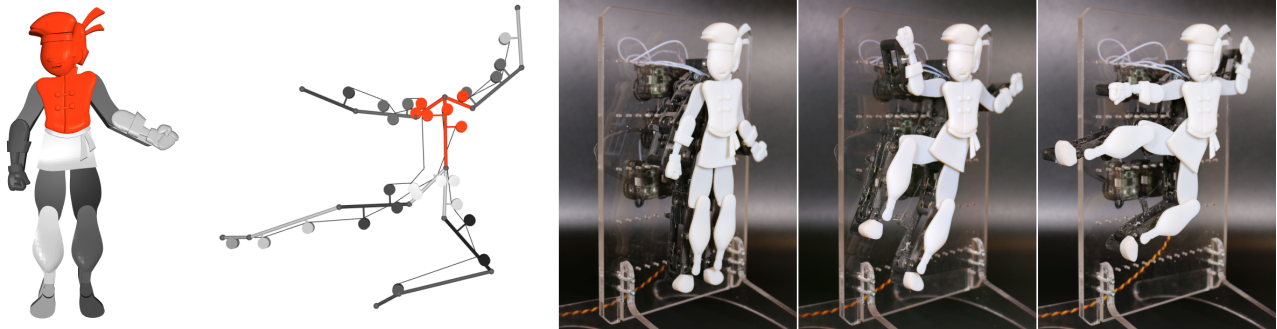


Figure 1: Our computational tool for designing cable-driven kinematic chains and trees (left) enables artists and hobbyists to size and place a cable network (middle) in order to closely match a set of target poses or keyframes using co-optimized control forces (right).

ABSTRACT

In this paper we present an optimization-based approach for the design of cable-driven kinematic chains and trees. Our system takes as input a hierarchical assembly consisting of rigid links jointed together with hinges. The user also specifies a set of target poses or keyframes using inverse kinematics. Our approach places torsional springs at the joints and computes a cable network that allows us to reproduce the specified target poses. We start with a large set of cables that have randomly chosen routing points and we gradually remove the redundancy. Then we refine the routing points taking into account the path between poses or keyframes in order to further reduce the number of cables and minimize required control forces. We propose a reduced coordinate formulation that links control forces to joint angles and routing points, enabling the co-optimization of a cable network together with the required actuation forces. We demonstrate the efficacy of our technique by designing and fabricating a cable-driven, animated character, an animatronic hand, and a specialized gripper.

* The first three authors contributed equally.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SCA '17, July 28-30, 2017, Los Angeles, CA, USA

© 2017 Copyright held by the owner/author(s). Publication rights licensed to Association for Computing Machinery.

ACM ISBN 978-1-4503-5091-4/17/07...\$15.00

<https://doi.org/10.1145/3099564.3099576>

CCS CONCEPTS

• **Computing methodologies** → **Animation**; *Physical simulation*;

ACM Reference format:

Vittorio Megaro*, Espen Knoop*, Andrew Spielberg*, David I.W. Levin, Wojciech Matusik, Markus Gross, Bernhard Thomaszewski, and Moritz Bächer. 2017. Designing Cable-Driven Actuation Networks for Kinematic Chains and Trees. In *Proceedings of SCA '17, Los Angeles, CA, USA, July 28-30, 2017*, 10 pages.

<https://doi.org/10.1145/3099564.3099576>

1 INTRODUCTION

Graphics research has 30 years of expertise in developing tools which allow digital artists to create expressive animations by posing a hierarchical set of rigid links. They breathe life into these articulated assemblies by making them move or locomote like a human, a familiar character, an animal or a fantasy creature. Tools like these enable artists to bring animated characters in feature film to life, giving them a unique personality.

With the advent of consumer-level digital fabrication technologies and powerful yet affordable off-the-shelf electronic components, artists now have the machinery at their disposal to make these articulated, animated assemblies physical. Creating such devices involves the design of a kinematic structure, determining the possible range of motion along with an actuation mechanism in order to animate the structure. Applications include animatronics, personalized robotics, and marionette design.

As the digital animations do not have to obey the laws of physics, such kinematic assemblies can often be slender and their motion fast. This makes it infeasible to place a motor at each joint due to both their size and weight. An alternative to distributed actuation is to use cables, placing actuators in a centralized location away from the mechanical assembly, enabling lightweight designs.

In this paper we present a method that designs a cable network, aiding the artist and hobbyist with the design of cable-driven kinematic chains and trees (see Fig. 1) that closely match a set of specified poses or keyframes when actuated. A theoretical upper bound on the number of required cables is two per rotational degree of freedom because we can only pull on cables. To reduce the cost and complexity of the kinematic assemblies, we wish to minimize the number of cables far beyond this upper bound. This presents a challenging design problem as it is of discrete nature and cables introduce non-trivial couplings and non-linearities. We place torsional springs at the joints to permit unidirectional actuation as cables can only exert pulling forces. We then co-optimize the number and placement of cables together with the control forces needed to drive the mechanical hierarchies.

We tackle the automated design with a two-step approach where we first identify the topology of a network by removing unactuated cables from a large set with routing points chosen at random. In a second step, we refine this network by parameterizing the routing points, taking the path between poses or keyframes into account, and further reducing the network and control forces if possible. To enable co-optimization of cable routing points and actuation forces, we introduce torque equilibrium equations that directly relate joint angles and routing points to the control forces.

The robotics community has proposed a myriad of cable-driven hands [Catalano et al. 2012; Grebenstein 2014; Ma et al. 2013] or full-bodied robots [Hannaford et al. 2013; Rooks 2006; Spröwitz et al. 2014]. However, designing these kinematic assemblies manually, roboticists focus on the optimal exchange of forces during physical interactions with humans or the environment. Our work is complementary to these techniques and targets an *automated* and *optimal* routing of a complex cable network under a no-contact assumption.

Our implementation is quasi-static and our physical assemblies are designed to meet this assumption. We validate our results by fabricating three of our optimized designs. With our examples, we illustrate applications in functional as well as artistic design. We show in the accompanying video that our kinematic chains and trees match the user-specified poses and keyframes well.

Contributions. In summary, we present

- the first algorithm to design artist-controlled, fabricable cable-driven kinematic chains and trees
- a reduced coordinate formulation that couples actuation forces to joint angles and routing points, enabling quasi-static simulation and co-optimization of routing points and control forces
- a two-step optimization approach to size and place a complex cable network to match a set of user-specified target poses or keyframes in a least squares sense

2 RELATED WORK

Our community has recently started to explore design tools for various mechanical systems, ranging from structurally-sound static objects [Dumas et al. 2015; Martínez et al. 2015; Prévost et al. 2013] to functional mechanical assemblies [Coros et al. 2013]. Our work is particularly inspired by recent research on designing articulated, mechanical characters [Bächer et al. 2012; Coros et al. 2013; Zhu et al. 2012]. Closest to our approach is the method by Coros et al. [2013], which uses a database of parameterized mechanisms with fixed topology in order to enable user-driven, interactive design of gear-based mechanical characters. Subsequent work [Thomaszewski et al. 2014] has proposed a user-guided method for transforming animated kinematic chains into multi-bar linkages that approximate the input motion with only a single motor. Other works have shown how to build gear-driven mechanical automata directly from motion capture sequences of human motion [Ceylan et al. 2013], rapidly craft linkage-based characters [Megaro et al. 2014], edit existing linkages while retaining their functionality [Bächer et al. 2015], and how to interactively design robotic creatures with desired morphologies and motions [Megaro et al. 2014].

Unlike these works, which focus on the design of articulated linkage or gear-driven characters, we instead focus on artist-controlled linkages that are actuated using cables routed through point-to-point connections spanning one or more joints in a linkage. The defining characteristic of this actuation paradigm is that only unidirectional actuation is possible – i.e. cables cannot *push* a link. Cable-driven designs have important design advantages over purely linkage- or gear-based approaches such as allowing significant control over the location of motor mass. For instance, a heavy motor can be located in the torso of a mechanical character while cables are used to actuate the limbs. This enables lightweight limbs that can therefore undergo more expressive motions. Furthermore, cables are easier to route than linkages meaning that they can more easily actuate several joints at once. Finally, because cables can span and couple multiple joints, cable-driven animatronic mechanisms may be able to better replicate the coupled motions inherent in many creatures [Clutterbuck and Jacobs 2010].

Given the desire to produce natural looking motions, computer graphics has actively explored the efficient simulation of cable-driven (also referred to as tendon-driven) systems [Sueda et al. 2011, 2008]. Furthermore, biomechanics literature has done extensive work in the efficient simulation of tendon-driven biomechanics (for instance OpenSIM [Delp et al. 2007]). There has recently been work on generating key-framed animations by applying both black-box and white-box control schemes to these systems [Sachdeva et al. 2015]. Our design system features a simulator for cable-driven mechanisms, but rather than previous fully dynamic simulations [Sachdeva et al. 2015; Sueda et al. 2011, 2008], we rely on a quasistatic assumption, allowing us to avoid costly time integration. While there has been work on fabricating cable-actuated folding surfaces [Kilian et al. 2017], this prior work focuses on folding origami shapes between open and closed positions, not the co-optimization of control and design for the motion of cable-driven linkages that we attack here. Co-optimization of design and control has been explored for building multirotors [Du et al. 2016] but not for the design of cable-driven mechanisms.

Finally, cable-driven mechanisms and biomechanical modeling have received much attention in robotics. However, many of the works in this field are targeted towards manually designing or learning controllers for specific mechanical designs such as spines [Mizuuchi et al. 2002], tentacles [Camarillo et al. 2008; Rucker and III 2011], arms, hands and fingers [Borghesan et al. 2010; Grebenstein 2014; Ma et al. 2013; Mitsui et al. 2013; Ozawa et al. 2009; Rooks 2006; Roy et al. 2013; Sawada and Ozawa 2012; Whitney et al. 2014] or parallel manipulators [Behzadipour 2005; Fang et al. 2004]. Surgical robots can also benefit from cable-driven actuation due to the fact that their end-effectors consist of small, surgical apparatuses [Hanaford et al. 2013]. The idea of a fixed design and optimizing for control parameters also extends to more esoteric designs such as continuum manipulators [Camarillo et al. 2008]. Control strategies have even been developed for full-body, cable-driven robots such as the ECCEROBOT [Potkonjak et al. 2011], RoBoy [Rob 2016] and Sparky [Spa 2016]. More recent work uses genetic algorithms to optimize cable tensions and cable angle to generate single, periodic trajectories for fixed, small DOF (2-3) designs [Bryson et al. 2016].

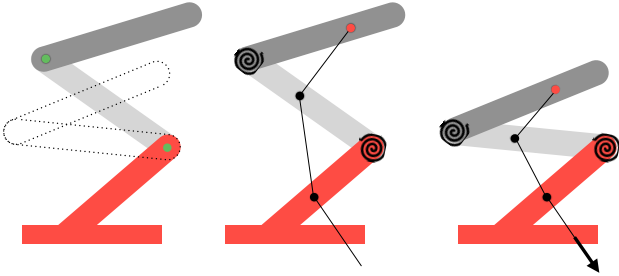


Figure 2: The input to our system is a kinematic assembly consisting of rigid links, jointed together at their ends (left). The red link is fixed. A user first specifies target poses (dotted link contours) and adds torsional springs to the hinges (green circles). A set of cables is then optimized (middle) to hit the user-specified target poses as closely as possible when actuated (right). The cables are attached at one end (red circle) and are routed with pulleys (black circles) to the fixed link. Elastic springs are added to the hinges (black spirals) to define the rest configuration as the zero energy state.

3 OBSERVATIONS

Before we formalize our cable-driven simulation and optimization approach on general hierarchical input, we discuss a series of observations on a three-link kinematic chain (see Fig. 2) that guide and motivate our representation and formulation.

As input we assume a *kinematic assembly* consisting of a single *kinematic chain* without loops, or a hierarchy of such chains to which we refer as *kinematic trees*. These assemblies consist of rigid links, connected to one another with mechanical hinges as illustrated in Fig. 2 left.

Given a kinematic assembly and one or several target poses (see dotted contour lines in Fig. 2 left), our goal is to determine a *cable network* (middle), i.e., a number of cables and corresponding routing points on the individual links of the assembly such that, when

applying specific forces to the cables, the assembly approximates the target poses as closely as possible (right).

Valid alternatives to a cable network are motors at the hinges or the addition of mechanical couplings between links [Thomaszewski et al. 2014]. A kinematic tree such as, e.g., a mechanical hand design, however, becomes bulky and heavy if a motor is added to each individual joint and inertial forces become prohibitively high. While leading to more lightweight designs, mechanical couplings between links are of fixed length and many of them are needed to achieve simple motions such as the contraction of a chain (compare Fig. 2 middle and right).

For a fully actuated kinematic tree, we would need twice the number of actuators (one to pull on either side of each hinge). However, although target poses generally involve non-zero deformations for all joints, we can exploit the inherent low-dimensionality of the problem. To this end, we first add elastic springs to the joints (see Fig. 2 middle). We then strategically place routing points of a cable network that leads to a complex coupling between individual joints and thus significantly reduces the number of cables (actuators) without compromising shape approximation. Note that the torsional springs uniquely define the rest configuration as the state of zero elastic energy.

Following the standard formulation described by Coros and colleagues [2013], we first experimented with representing the state of each rigid link with a position and orientation, and hinges and cables interconnecting them with non-linear constraints. However, the non-negativity of actuation forces (we can only pull on cables) require additional inequality constraints on the cable lengths that are non-trivial (refer to Fig. 6 for a small example): if we pull on one cable with a non-zero force other cables may extend in length while they remain unactuated.

Departing from this full coordinate formulation, we introduce torque equilibrium equations that directly relate joint angles and routing points to the control forces, avoiding any non-trivial inequality constraints.

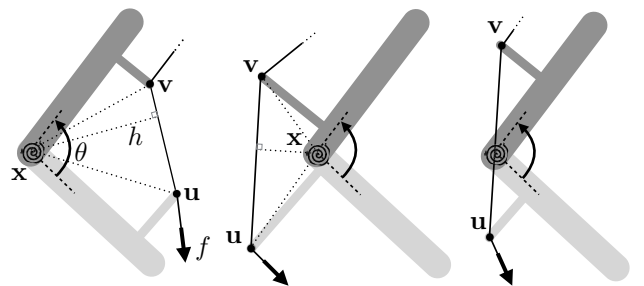


Figure 3: The assembly is in equilibrium if the torque $\tau_s(\theta)$ of the torsional spring with joint angle θ equals the applied torque hf with signed moment arm $h = \frac{\det[\mathbf{u}-\mathbf{x}, \mathbf{v}-\mathbf{x}]}{\|\mathbf{u}-\mathbf{v}\|}$.

4 SIMULATING CABLE-DRIVEN TREES

We base our formulation on the following observation: if we pull on the cable in Fig. 3 right with a force $f > 0$, we apply a torque hf about \mathbf{x} where h is the moment arm of the oriented triangle

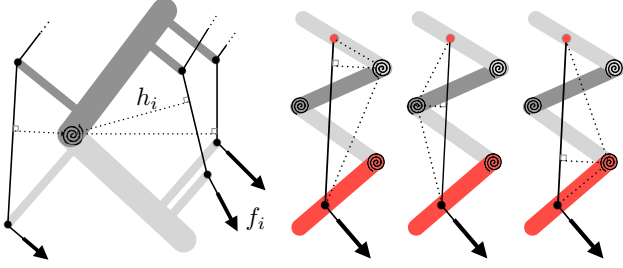


Figure 4: Torques $h_i f_i$ from several cables can affect the position of a single joint (left) and a single cable can affect the positions of several joints (right).

$(\mathbf{x}, \mathbf{u}, \mathbf{v})$. The system is in equilibrium if this torque equals the one from the torsional spring $\tau_s(\theta)$ that evaluates to a positive value if the oriented joint angle θ is smaller than the angle at rest. This observation holds if the cable is attached to respective links with friction- and dimensionless pulleys (black circles). We defer a discussion of finite-dimensional pulleys to Sec. 6.

To relate the moment arm to the joint location \mathbf{x} and the two routing points \mathbf{u} and \mathbf{v} , we express the signed triangle area with Cramer's rule $\frac{1}{2} \det[\mathbf{u} - \mathbf{x}, \mathbf{v} - \mathbf{x}]$ and set it equal to half the triangle's altitude h times its base $\|\mathbf{u} - \mathbf{v}\|$.

This equilibrium condition still holds if the triangle orientation is flipped as illustrated in Fig. 3 middle: if we pull on the cable, the oriented angle θ becomes larger than the one at rest. Hence, the torsional spring torque becomes negative. Due to the change of orientation of the triangle $(\mathbf{x}, \mathbf{u}, \mathbf{v})$, $\det[\mathbf{u} - \mathbf{x}, \mathbf{v} - \mathbf{x}]$ is negative and compensates this sign change. A special case is depicted in Fig. 3 right: if the triangle area becomes zero ($h = 0$), the applied torque is zero independent of the magnitude of the applied force. Hence, the elastic spring remains in its zero energy state for all $f > 0$. However, such configurations are unstable equilibria and will therefore not be present in real systems.

The above observation is pivotal when we optimize routing points: if we move the routing points from one side to the other (flip of triangle orientation), the moment arm h changes smoothly and the behavior of the cable network is well defined.

For a hierarchical input assembly, we formulate an equilibrium condition for each individual mechanical hinge j . If several cables i exert torques at j (refer to Fig. 4 left), we sum up their contributions

$$\tau_j(\theta_j) = \tau_s(\theta_j) - \sum_i h_{ij} f_i = 0 \quad (1)$$

where the elastic spring behavior $\tau_s(\theta_j)$ may vary from joint to joint and its stiffness could be non-linear.

To determine the torque a cable i exerts on j (if any), we seek the first routing points \mathbf{u}_{ij} and \mathbf{v}_{ij} on the paths from j to the root and the leaves, respectively (compare with Fig. 4 right). If only one routing point is found, no torque is exerted. Otherwise, the moment arm

$$h_{ij} = \frac{\det[\mathbf{u}_{ij} - \mathbf{x}_j, \mathbf{v}_{ij} - \mathbf{x}_j]}{\|\mathbf{u}_{ij} - \mathbf{v}_{ij}\|} \quad (2)$$

is computed and the torque $h_{ij} f_i$ added to equation j .

Collecting all actuation forces f_i in a vector \mathbf{f} and the per-joint angles θ_j in a vector $\boldsymbol{\theta}$, we can then find the equilibrium state $\boldsymbol{\theta}$ for a given \mathbf{f} by solving the non-linear torque equations

$$\boldsymbol{\tau}(\boldsymbol{\theta}) = 0 \quad (3)$$

using the Levenberg-Marquardt algorithm.

It remains to discuss how we express the global joint and node locations (in brown in Fig. 5 left) with joint angles and the routing points w.r.t. these node locations (right).

As aforementioned, we assume our input to be hierarchical and therefore loop-free. Hence, we rely on a recursive definition as commonly used for rigs in character animation: the topology of the hierarchy is uniquely defined with a function r that maps a node k to its respective parent $r(k)$ (see Fig. 5 left). The root node is mapped to the origin o . To transform the position \mathbf{x}^k in frame k to the frame of its parent $r(k)$, we first apply a rigid transformation with constant rotation $\mathbf{R}^{k \rightarrow r(k)}$ and translation $\mathbf{t}^{k \rightarrow r(k)}$, then rotate counterclockwise by angle θ_j if the parent is a mechanical hinge j

$$\mathbf{x}^{r(k)} = \mathbf{R}^{r(k)} \left(\mathbf{R}^{k \rightarrow r(k)} \mathbf{x}^k + \mathbf{t}^{k \rightarrow r(k)} \right)$$

where the rotation $\mathbf{R}^{r(k)}$ is either

$$\begin{bmatrix} \cos(\theta_j) & -\sin(\theta_j) \\ \sin(\theta_j) & \cos(\theta_j) \end{bmatrix}$$

if $r(k)$ equals a joint j and the identity otherwise. Using this rule recursively, we transform positions from local to the world frame $\mathbf{x}^o(\boldsymbol{\theta})$, omitting the superfix o for positions in global coordinates. Note how nodes that move rigidly with a link depend on all joint angles θ_j on the path from this link's frame to the root. Hence, leaves depend on more angles than nodes closer to the root, leading to sparsity differences in derivatives of $\mathbf{x}(\boldsymbol{\theta})$ w.r.t. the joint angles.

To rigidly move the routing points with their respective links, we define local, per-link-segment frames (compare with Fig. 5 right) where the difference vector \mathbf{d} , pointing from parent to child node, and its perpendicular vector \mathbf{d}^\perp define the local link frame uniquely. Routing points \mathbf{u} and \mathbf{v} are then computed according to

$$\mathbf{x} + p^\perp \mathbf{d}^\perp + p \mathbf{d} \quad \text{with} \quad \mathbf{d} = \begin{bmatrix} d_x \\ d_y \end{bmatrix} \quad \text{and} \quad \mathbf{d}^\perp = \begin{bmatrix} d_y \\ -d_x \end{bmatrix}$$

where \mathbf{x} is the node position of the parent. Note that \mathbf{d}^\perp stays on the right of \mathbf{d} , independent of the link's orientation.

While we keep the coordinates of these routing points constant during simulations, we optimize their number and placement to closely match user-specified target poses. To this end, we collect all coordinates p^\perp and p in a parameter vector \mathbf{p} .

5 PLACING AND SIZING CABLE NETWORKS

Before optimizing a cable network, a user defines target poses or a sequence of keyframes t by specifying desired locations $\tilde{\mathbf{x}}_s$ for a subset of the tree nodes as we illustrate in the accompanying video. We then use standard inverse kinematics (IK)

$$\min_{\boldsymbol{\theta}} \sum_s \|\mathbf{x}_s(\boldsymbol{\theta}) - \tilde{\mathbf{x}}_s\|^2 + \gamma \|\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{prev}}\|^2$$

to solve for the target angles $\boldsymbol{\theta}^t$ where we add a regularization term that keeps the current solution close to the previous one, controlled by a weight γ .

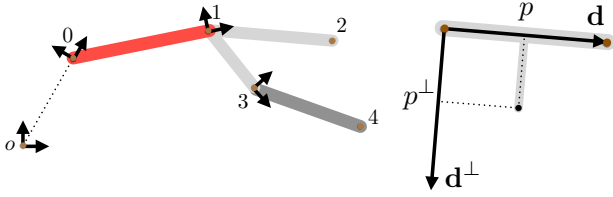


Figure 5: To describe the kinematics of our hierarchical input, we use a recursive definition similar to the one used for rigs in character animation (left): this example consists of two hinges (at nodes 1 and 3) and three components (fixed link in red, link consisting of two segments in light grey, “leaf” link in dark grey). The topology is uniquely defined by the function $r = \{(0, o), (1, 0), (3, 1), (4, 3), (2, 1)\}$. We express routing points in local, per-segment frames $[d^\perp, d]$ with coordinates (p^\perp, p) (right).

5.1 Identifying the Network Topology

Given a target pose θ^t , we can compute the torsional spring torques $b_j^t = \tau_s(\theta_j^t)$ that require compensation, directly. And if we temporarily assume for a moment that the cables with their routing points \mathbf{p} are known, we can compute the moment arms h_{ij} from the nodal positions $\mathbf{x}(\theta^t)$. Hence, the equilibrium equations 1 become linear in the unknown actuation forces

$$\mathbf{H}^t \mathbf{f}^t = \mathbf{b}^t.$$

This observation paves the way for a simple heuristic to identify the topology of a cable network that is able to hit the target poses exactly: we generate many cables (more than a thousand) of varying “lengths” by randomly choosing the number and coordinates p^\perp and p of their routing points in user-provided ranges smaller or equal to $[-1, 1]$ for p^\perp and $[0, 1]$ for p . In other words, we randomly fill in rows of the \mathbf{H}^t matrices, rendering these equation systems highly underdetermined. And because the probability of choosing two linear dependent rows at random is practically zero, the underdetermined, per-target systems can be solved exactly even if we constrain the forces \mathbf{f}^t to be positive.

Such a cable network is highly impractical though. As previously mentioned, for a fully actuated assembly two cables per mechanical hinge – one on either side – are sufficient. Given a set of target poses, we aim at finding a network with far fewer cables than this upper bound.

To this end, we formulate a sparsity regularizer R_{sparse} that favors a cable to remain unactuated across all targets and solve the resulting constrained optimization problem

$$\min_{\mathbf{f}^t} R_{\text{sparse}}(\mathbf{f}^t) \quad \text{s.t.} \quad \mathbf{H}^t \mathbf{f}^t - \mathbf{b}^t = 0 \quad \text{and} \quad \mathbf{f}^t > 0 \quad (4)$$

where we bound the forces to only be pulled on.

For our sparsity regularizer, we approximate the $L1$ -norm similar to Skouras et al. [2013]

$$R_{\text{sparse}}(\mathbf{f}^t) = \sum_i \left(\sum_t f_i^t \right)^\alpha$$

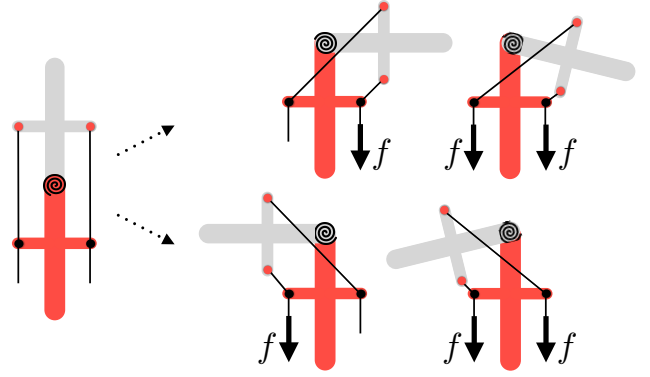


Figure 6: Dependent on the sequence of actuation, the symmetric single-joint assembly with two cables (left) ends up in two completely different configurations: the force f is first applied to the right, then to the left cable (top row). If this sequence is reversed (bottom row), the assembly tilts to the left instead.

where index i runs over all cables and t over all targets. α is set to a fraction smaller than 1. We use $\alpha = 0.3$ for all our results.

If a cable remains unactuated for all targets, we remove it from the initial set, resulting in a small cable network with routing points \mathbf{p} and actuation forces \mathbf{f}^t .

5.2 Refining the Cable Network

When solving for sparse actuation forces by minimizing Problem 4, we ignore the path from the unactuated configuration to a particular target or, alternatively, from one keyframe to another. However, in general, the end configuration depends on the sequencing of actuations as we illustrate in Fig. 6 with a two-cable example. Only if none of the moment arms h_{ij} flips its sign under active cables i is the end configuration independent of the sequence of actuation. We experimented with complementarity constraints to enforce path independence. Disallowing sign changes, however, is too restrictive and leads to networks with more cables than necessary.

To take path dependence into account, we simulate from unactuated to target configurations or from keyframe to keyframe each time we evaluate our refinement objective, constraints, and their derivatives. We thereby ensure that the static equilibrium 3 can be reached from the respective starting point.

To keep the actuated assembly close to the target poses during co-optimization of \mathbf{p} and \mathbf{f}^t , we formulate a target matching objective

$$g_{\text{target}}(\mathbf{p}, \mathbf{f}^t) = \sum_t \sum_k \|\mathbf{x}_k(\mathbf{p}, \mathbf{f}^t) - \tilde{\mathbf{x}}_k^t\|^2$$

where index k runs over the nodes of the assembly and $\mathbf{x}_k^t = \mathbf{x}_k(\theta^t)$.

A second goal of our refinement optimization is the sizing of motors where we strive for the weakest actuation necessary, and the removal of additional cables if possible. To this end, we add an *actuation* regularizer

$$R_{\text{actuation}}(\mathbf{f}^t) = \gamma_n \sum_t \|\mathbf{f}^t\|^2 + \gamma_s R_{\text{sparse}}(\mathbf{f}^t)$$

with two relative weights γ_n and γ_s to regulate their influence.

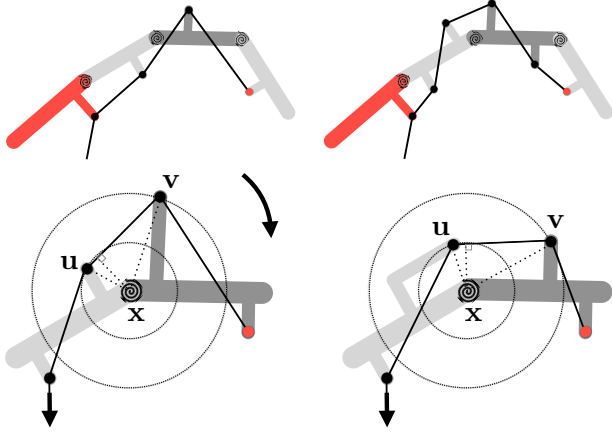


Figure 7: We use two instead of a single routing point per cable and link (top row). While this adds to the complexity of the manual assembly task (the pulleys almost double per cable), it has an important advantage when generating geometry for the final assemblies: we can rotate oriented triangles (x, u, v) about the hinge axis without changing the torque equilibrium in any given pose.

Minimizing actuation forces alone, however, renders the optimization problem unbound because we can compensate a smaller force f by increasing the moment arm h . Hence, we constrain the routing points to stay within reasonable bounds away from their rigid links $\mathbf{p} \in [\mathbf{p}_{\text{lower}}, \mathbf{p}_{\text{upper}}]$.

To avoid inconsistencies in the angles due to the periodicity of the sines and cosines defining our joint rotations, we add an additional regularizer that keeps the rotation angles close to the user-specified ones, normalized to the range $[0, 2\pi]$

$$R_{\text{period}}(\mathbf{p}, \mathbf{f}^t) = \gamma_p \sum_t \|\theta(\mathbf{p}, \mathbf{f}^t) - \theta^t\|^2$$

with a weight γ_p .

Eyeballing the expression for our signed moment arm (see Eq. 2), we divide by zero if the routing points \mathbf{u}_{ij} and \mathbf{v}_{ij} adjacent to joint j collapse to a single point. To prevent close proximity between neighboring routing points, we add inequality constraints on their distance. Deferring a detailed discussion of finite-dimensional pulleys to the next section, these constraints safeguard against pulley-pulley collisions and we add similar constraints to keep joints and pulleys a safe distance away from one another.

In summary, we solve the constrained co-optimization problem

$$\begin{aligned} \min_{\mathbf{p}, \mathbf{f}^t} \quad & g_{\text{target}}(\mathbf{p}, \mathbf{f}^t) + R_{\text{actuation}}(\mathbf{f}^t) + R_{\text{period}}(\mathbf{p}, \mathbf{f}^t) \\ \text{s.t.} \quad & \forall i, j : \|\mathbf{u}_{ij}(\mathbf{p}, \mathbf{f}^t) - \mathbf{v}_{ij}(\mathbf{p}, \mathbf{f}^t)\| > 2r_p \\ & \|\mathbf{u}_{ij}(\mathbf{p}, \mathbf{f}^t) - \mathbf{x}_j(\mathbf{p}, \mathbf{f}^t)\| > r_p + r_h \\ & \|\mathbf{v}_{ij}(\mathbf{p}, \mathbf{f}^t) - \mathbf{x}_j(\mathbf{p}, \mathbf{f}^t)\| > r_p + r_h \\ & \text{and } \mathbf{p}_{\text{lower}} < \mathbf{p} < \mathbf{p}_{\text{upper}} \end{aligned}$$

with r_p and r_h referring to pulley and mechanical hinge radii, respectively.

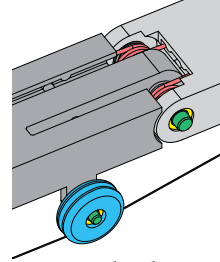
To avoid coupling between neighboring moment arms h_{ij} (see Fig. 7), we use two instead of a single routing point per rigid link (top row). While these almost double the number of pulleys per cable, it has an important advantage when generating the geometry for our output assemblies (bottom row, see Sec. 6): when rotating the oriented triangle (x, u, v) about the hinge axis, the moment arm and resulting torque remains the same, independent of the pose we are in. To favor a particular solution in these joint-cable subspaces, one can add a weak regularizer on the routing points. Note that the coupling due to the constant force magnitude along the cable remains.

To compute gradients, we first simulate to equilibrium $\tau(\theta) = 0$. Collecting the routing points and per-target actuations $\mathbf{q} = (\mathbf{p}, \mathbf{f}^t)$, we then use the implicit function theorem

$$\frac{d\tau(\mathbf{q}, \theta(\mathbf{q}))}{d\mathbf{q}} = \frac{\partial\tau(\mathbf{q}, \theta)}{\partial\mathbf{q}} + \frac{\partial\tau(\mathbf{q}, \theta)}{\partial\theta} \frac{\partial\theta(\mathbf{q})}{\partial\mathbf{q}} = 0$$

to compute the analytical gradients $\frac{\partial\theta(\mathbf{q})}{\partial\mathbf{q}}$. Remaining derivatives are computed at runtime using symbolic differentiation [Gunter 2007].

6 FABRICATION CONSIDERATIONS



To fabricate our cable-driven kinematic chains and trees, we use a combination of 3D printing and standard off-the-shelf parts as illustrated in the inset on the left with a computer-aided design (CAD) drawing: our mechanical hinges consist of a standard helical torsion spring (in red), a shaft (in green), and a bearing (in yellow) at either end of the shaft. For the routing points we use a similar design with a pulley (in blue) and for our cables (in black) fishing line.

use a similar design with a pulley (in blue) and for our cables (in black) fishing line.

Friction. It is of paramount importance to reduce friction at the hinges and pulleys to a minimum and bearings serve this purpose. However, in physical kinematic assemblies friction is unavoidable and as a rule of thumb, friction in pulleys leads to a decrease in tension along the cable from the root to the leaves of the hierarchy. At mechanical joints, we may observe several equilibrium states in a local neighborhood.

Torsional Springs. Our standard torsional springs are labeled as linear and symmetric. However, a characterization experiment reveals a linear but asymmetric behavior (see Fig. 8) even after oiling the springs to avoid friction between spring windings. To avoid a C^0 spring torque, we use a sigmoid function

$$\tau_s(\theta) = s(\theta)k_{\text{pos}} + (1 - s(\theta))k_{\text{neg}} \quad s(\theta) = \frac{1}{1 + e^{-\beta\theta}}$$

in our simulation and co-optimization where we set β to 100.

Spring Stiffness Range. When choosing a stiffness range for our torsional springs, we strive for balancing the trade-off between meeting our quasi-static assumption (lower stiffness bound) and the maximum actuation forces (upper bound). If springs are too soft, a quasistatic simulation is insufficient due to the non-negligible

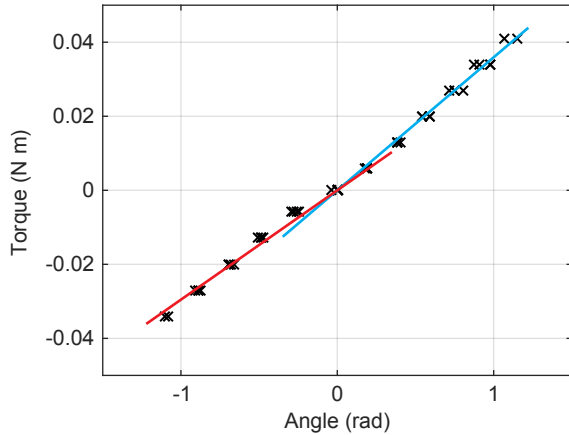


Figure 8: A characterization experiment reveals a linear but asymmetric behavior with k_{pos} in blue and k_{neg} in red.

effect of inertia. If the control forces are too high, actuation by hand or with inexpensive, low-end servos is infeasible.

A dynamic simulation of our cable-driven kinematic assemblies is straightforward

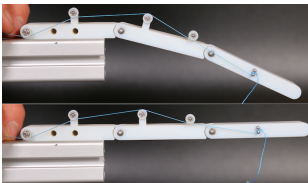
$$I \frac{d^2 \theta_j}{dt^2} + C \frac{d\theta_j}{dt} + \tau_s(\theta_j) = \sum_i h_{ij} f_i$$

with moment of inertia I and damping C . However, the corresponding cable network optimization is difficult as it requires a space-time formulation [Witkin and Kass 1988] where error may accumulate over time.

Because of the use of bearings at joints and pulleys, the damping C in our system is small. Hence, the frequency of vibration at a joint is expected to be close to the natural resonance frequency

$$f_n = \frac{1}{2\pi} \sqrt{\frac{k}{I}}$$

where k is the linearized spring stiffness. Given a lower bound on the frequency and an estimate of the moment of inertia, we can use above relation to compute a first order approximation of the lower bound of the stiffness range that renders a quasi-static assumption valid. An upper bound can be computed from a desired maximum force magnitude.



Gravity Compensation. Due to this upper bound on stiffness, our assemblies sag under gravity and require compensation as illustrated in the inset on the left: the initially straight, unactuated three-link assembly “bends” under gravity (top) and only if accounted for, we can hit the target of a straight, actuated assembly (bottom). To compensate for gravity at a joint \mathbf{x}_j (see Fig. 9), we keep all other torsional springs fixed, transform centers of mass \mathbf{c} of child links, joints, and pulleys from

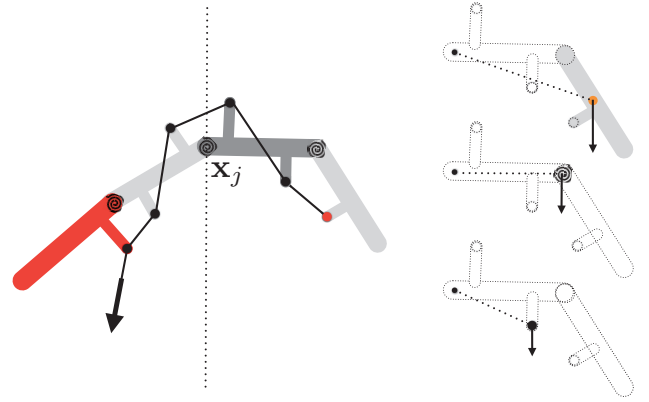


Figure 9: We compensate for gravity at a particular joint \mathbf{x}_j by holding all other torsional springs fixed (left), adding up gravitational torques of rigid links (right, top), torsional springs (right, middle), and pulleys (right, bottom) on the path from j to the leaves.

local to global coordinates, then add their gravitational torques

$$(\mathbf{c} - \mathbf{x}_j) \times \begin{bmatrix} 0 \\ -mg \end{bmatrix}$$

to the equilibrium equation 1 for joint j . W.l.o.g. we assume gravity to point in the negative y direction. m is the mass of the respective link, joint, or pulley and g is the standard gravity. During network topology identification (see Eq. 4), we ignore gravitational effects of pulleys as the randomly chosen cables with their many routing points would lead to a massive weight, rendering the optimization unrealistic. During our refinement, gravity compensation is fully switched on.

Finite-Dimensional Pulleys. Pulleys are not dimensionless and we cannot route cables through their centers. As shown in Fig. 10 we account for their finite dimension by adding an offset r_p to all moment arms h_{ij} during simulations and optimization where r_p is the radius of the pulley. Note that this offset is – independent of the pose – of constant value r_p as the cable is perpendicular to the h_{ij} s and the pulleys are circular. As illustrated in Fig. 10 in blue, we wrap cables once around the pulleys to ensure that they do not detach during actuations. We do that in a consistent manner for all our pulleys.

Generating Geometry. For the database of supported springs, we manually generate negative and positive hinge geometry that supports mounting of respective springs by snapping them in place. The joint geometry is instantiated and remaining geometry automatically generated by our system. If a routing point collides with a rigid link other than the one it corresponds to, the user rotates the oriented triangle to avoid these collisions as illustrated in Fig. 7 bottom.

7 RESULTS

We have used our computational tool to size and place cable networks for a total of three examples: a gripper (Fig. 12), an animatronic hand (Fig. 13), and an animated fighter character (Fig. 1).

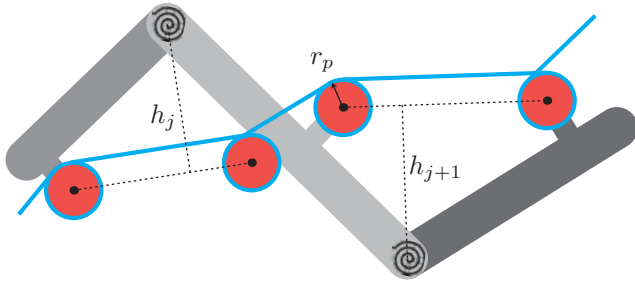


Figure 10: We account for the finite dimension of pulleys by offsetting all moment arms h_j and h_{j+1} by the constant pulley radius r_p . To ensure that cables do not detach during actuation, we wrap them once around the pulleys (see cable in blue).

We refer to the accompanying video for footage of the physical assemblies in action. We summarize key features and complexity of the physical assembly in Tab. 1.

Model	links	joints	targets	cables	pulleys
Fighter	12	11	2	7 (4000, 12)	31
Hand	13	12	4	4 (3500, 12)	26
Gripper	13	12	2	4 (6000, 18)	48

Table 1: We summarize the complexity (number of links, joints, specified targets, cables, and pulleys) of our three example assemblies. Our two-step optimization reduces the initial number of cables x to y after the first, then to z after the second step, formatted in column “cables” using $z(x, y)$.

Fabrication. The rigid links for all our examples were printed on an Object Connex 350 with Vero White and Vero Black. After printing and cleaning, manual assembly of a model takes a few hours. We use double torsional springs (parts 8548 and 8555, Lesjofors AB), with tabulated spring stiffnesses of 0.0484 and $0.0950 \text{ Nm rad}^{-1}$, respectively. Pins and bearings are press fitted, speeding up the assembly process.

Validation. We validated our cable network optimization on the lower body part of our Fighter character seen in Teaser 1. It is a kinematic tree with multiple cables acting on the same link and an example that requires gravity compensation. We built a setup to replicate the optimized actuation forces experimentally. To this end, we use standard Newton meters pulling on the three ends of the cables as we illustrate in the accompanying video. Excursions are adjusted manually in order to match the prescribed forces. As we show in Fig. 11, we validate the performance of our target-matching objective on two target poses specified by the user. While we observe a slight mismatch for the lower knee joint of the right leg (bottom, right in Fig. 11), it can be seen that simulated and physical poses match well. This validates our modeling assumptions.



Figure 11: We validate our cable network optimization on the lower body of our Fighter character on two target poses (bottom, top) by controlling the cable forces with standard Newton meters. The simulated poses (left) match the physical poses (right) well.

Performance. For the lower body of the Fighter, we chose 1600 cables at random, then used our topology identification (Eq. 4) to reduce it the number of cables to 8 in 25 seconds. We then use our refinement to further reduce the number of cables to a total of 3. Our refinement takes 181 seconds on this example. Note that for hierarchies with several branches attached to the fixed link (compare with skeleton in our teaser), we optimize each branch independently. The lower body of the Fighter is the most complex branch with regards to time complexity, hence, representative.

Gripper. As a functional example, we designed a gripper to be able to pick up two kinds of differently shaped objects: in one configuration we can pick up two T-shaped objects, in the other a heart-shape. No gravity compensation is needed as the gripper operates in the horizontal plane. The gripper is symmetric and uses two cables on either side. As the gripper is operated manually, we wish to reduce the control complexity and optimize one cable per target pose and side.

Hand. Increasing the complexity, we optimize an animatronic hand with three fingers and a thumb as well as an actuated wrist. Note that the thumb is operated on a plane that differs from the horizontal one, hence, requiring gravity compensation where we work with projected gravity. The closing motion of each finger is optimized to follow four keyframes as can be seen in the supplemental video. Actuation of the middle fingers is coupled to movement of the wrist (fixed link) to yield a more realistic closing of the hand.

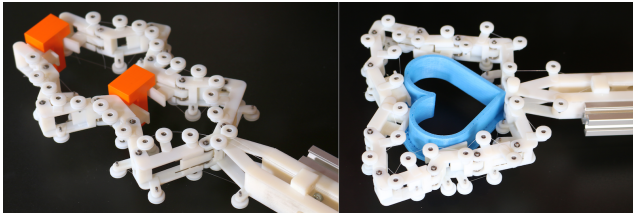


Figure 12: Our gripper is optimized to pick up two T-shapes when one cable is actuated, and a heart-shape if only the other is actuated.

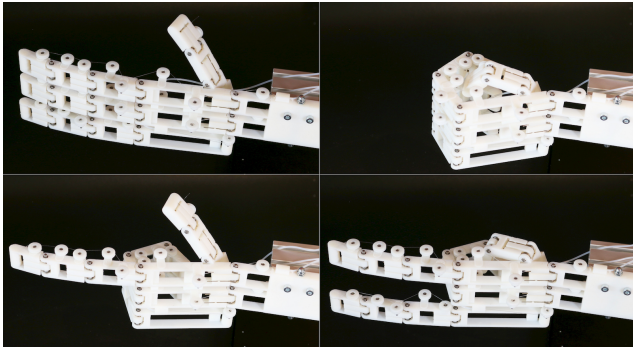
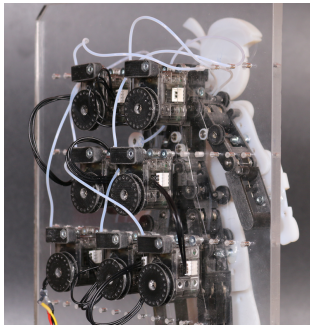


Figure 13: Our animatronic hand can perform a wide range of gestures. Its thumb is operating in a different plane than the remaining fingers and the wrist.

The cables of the thumb and pinky end on the palm and we use Bowden cables between the palm and the wrist to avoid exertion of torques on the joint connecting wrist with palm. The hand assembly can perform a wide range of gestures as can be see in Fig. 13 as well as the video. This example illustrates that our approach extends to the third dimension by allowing joints to lie in different planes.



Fighter. Our technique can be used to create animated, mechanical characters by actuating the cables with servos. Our Fighter is driven by a total of seven servos as shown in the inset on the left. We use Dynamixel XL-320 servos, with a stall torque of 0.39 Nm. The servos are located outside the character allowing for a lightweight design and we again use Bowden

cable to connect cables to servos. Note that the Bowden cables lead to non-negligible friction. As the servos are not force controlled, we adjust the excursions of the servos to match the user-specified keyframes.

8 CONCLUSION

We have devised a method to design complex cable networks for animating mechanical characters or controlling the motion of functional assemblies. We have introduced a reduced coordinate formulation that links joint angles and routing points to control forces, enabling co-optimization of routing and actuation. To identify the topology of a cable network, we route a large set of cables from root to leaves, attaching them to rigid links with routing points chosen at random, then removing redundancy. We further refine these networks, taking path dependence into account.

Limitations. For our gripper design, we assume contact forces to be negligible, an assumption not holding true when picking up objects of considerable weight. When in contact with the lightweight T-shapes or the heart, we do observe deviations from the intended gripping poses if we pull the cables beyond the optimized actuations. External forces lead to additional, pose-dependent torques that we could compensate for by adding them to the equilibrium equations as we did for the gravitational torques.

Future Work. There are several challenges remaining. Our technique assumes hierarchical input and an extension to mechanisms with loops is left as future work. We further plan to extend our technique to spatial input, adding support for joints with more than a single degree of freedom. As another interesting direction, we would like to explore optimization of a cable-network for assemblies where the source of compliance is not added by springs but a foam or silicone skin instead. Note that our spring torques allow for non-linear behavior as would be expected when using foam or silicone. Relaxing our assumption on quasi-statics opens another interesting avenue, requiring space-time optimization to size and place cable networks to control the dynamic behavior of assemblies.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful comments; Philipp Herholz for early explorations; Alessia Marra for model design. This work has been supported by the SOMA project (European Commission, Horizon 2020 Framework Programme, H2020-ICT-645599).

REFERENCES

- 2016. RoBoy. <http://roboy.org>. (2016). Accessed: 2016-12-28.
- 2016. Sparky. <http://rasc.usc.edu/sparky.html>. (2016). Accessed: 2016-12-28.
- Moritz Bächer, Bernd Bickel, Doug L. James, and Hanspeter Pfister. 2012. Fabricating Articulated Characters from Skinned Meshes. *ACM Trans. Graph.* 31, 4, Article 47 (July 2012), 9 pages.
- Moritz Bächer, Stelian Coros, and Bernhard Thomaszewski. 2015. LinkEdit: Interactive Linkage Editing Using Symbolic Kinematics. *ACM Trans. Graph.* 34, 4, Article 99 (July 2015), 8 pages.
- Saeed Behzadipour. 2005. *Ultra-high-speed Cable-based Robots*. Ph.D. Dissertation. University of Waterloo.
- G. Borghesan, G. Palli, and C. Melchiorri. 2010. Design of tendon-driven robotic fingers: Modeling and control issues. In *2010 IEEE International Conference on Robotics and Automation*. 793–798.
- Joshua T Bryson, Xin Jin, and Sunil K Agrawal. 2016. Configuration Robustness Analysis of the Optimal Design of Cable-Driven Manipulators. *Journal of Mechanisms and Robotics* 8, 6 (Dec. 2016), 061006.
- D. B. Camarillo, C. F. Milne, C. R. Carlson, M. R. Zinn, and J. K. Salisbury. 2008. Mechanics Modeling of Tendon-Driven Continuum Manipulators. *IEEE Transactions on Robotics* 24, 6 (Dec 2008), 1262–1273.
- M. G. Catalano, G. Grioli, A. Serio, E. Farnioli, C. Piazza, and A. Bicchi. 2012. Adaptive synergies for a humanoid robot hand. In *2012 12th IEEE-RAS International Conference*

- on *Humanoid Robots (Humanoids 2012)*. 7–14. <https://doi.org/10.1109/HUMANOIDS.2012.6651492>
- Duygu Ceylan, Wilnot Li, Niloy J. Mitra, Maneesh Agrawala, and Mark Pauly. 2013. Designing and Fabricating Mechanical Automata from Mocap Sequences. *ACM Trans. Graph.* 32, 6, Article 186 (Nov. 2013), 11 pages.
- S Clutterbuck and J Jacobs. 2010. A physically based approach to virtual character deformations. In *ACM SIGGRAPH*.
- Stelian Coros, Bernhard Thomaszewski, Gioacchino Noris, Shinjiro Sueda, Moira Forberg, Robert W. Sumner, Wojciech Matusik, and Bernd Bickel. 2013. Computational Design of Mechanical Characters. *ACM Trans. Graph.* 32, 4, Article 83 (July 2013), 12 pages.
- S. L. Delp, F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, E. Guendelman, and D. G. Thelen. 2007. OpenSim: Open-Source Software to Create and Analyze Dynamic Simulations of Movement. *IEEE Transactions on Biomedical Engineering* 54, 11 (Nov 2007), 1940–1950.
- Tao Du, Adriana Schulz, Bo Zhu, Bernd Bickel, and Wojciech Matusik. 2016. Computational Multicopter Design. *ACM Trans. Graph.* 35, 6, Article 227 (Nov. 2016), 10 pages.
- J r mie Dumas, An Lu, Sylvain Lefebvre, Jun Wu, and Christian Dick. 2015. By-Example Synthesis of Structurally Sound Patterns. *ACM Trans. Graph.* 34, 4 (July 2015), 12.
- Shiqing Fang, D. Franitz, M. Torlo, F. Bekes, and M. Hiller. 2004. Motion control of a tendon-based parallel manipulator using optimal tension distribution. *IEEE/ASME Transactions on Mechatronics* 9, 3 (Sept 2004), 561–568.
- Markus Grebenstein. 2014. *Approaching Human Performance*. Springer International Publishing.
- Brian Guenter. 2007. Efficient Symbolic Differentiation for Graphics Applications. *ACM Trans. Graph.* 26, 3, Article 108 (July 2007). <https://doi.org/10.1145/1276377.1276512>
- B. Hannaford, J. Rosen, D. W. Friedman, H. King, P. Roan, L. Cheng, D. Glozman, J. Ma, S. N. Kosari, and L. White. 2013. Raven-II: An Open Platform for Surgical Robotics Research. *IEEE Transactions on Biomedical Engineering* 60, 4 (April 2013), 954–959. <https://doi.org/10.1109/TBME.2012.2228858>
- Martin Kilian, Aron Monzpart, and Niloy J. Mitra. 2017. String Actuated Curved Folded Surfaces. *ACM Transactions on Graphics (to appear)* (2017).
- R. R. Ma, L. U. Odhner, and A. M. Dollar. 2013. A modular, open-source 3D printed underactuated hand. In *2013 IEEE International Conference on Robotics and Automation*. 2737–2743.
- Jon s Mart nez, J r mie Dumas, Sylvain Lefebvre, and Li-Yi Wei. 2015. Structure and Appearance Optimization for Controllable Shape Design. *ACM Trans. Graph.* 34, 6, Article 229 (Oct. 2015), 11 pages.
- Vittorio Megaro, Bernhard Thomaszewski, Damien Gauge, Eitan Grinspun, Stelian Coros, and Markus Gross. 2014. ChaCra: An Interactive Design System for Rapid Character Crafting. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '14)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 123–130.
- K. Mitsui, R. Ozawa, and T. Kou. 2013. An under-actuated robotic hand for multiple grasps. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 5475–5480.
- I. Mizuuchi, R. Tajima, T. Yoshikai, D. Sato, K. Nagashima, M. Inaba, Y. Kuniyoshi, and H. Inoue. 2002. The design and control of the flexible spine of a fully tendon-driven humanoid “Kenta”. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 3. 2527–2532 vol.3.
- R. Ozawa, K. Hashirii, and H. Kobayashi. 2009. Design and control of underactuated tendon-driven mechanisms. In *2009 IEEE International Conference on Robotics and Automation*. 1522–1527.
- Veljko Potkonjak, Bratislav Svetozarevic, Kosta Jovanovic, and Owen Holland. 2011. The Puller-Follower Control of Compliant and Noncompliant Antagonistic Tendon Drives in Robotic Systems. *International Journal of Advanced Robotic Systems* 8, 5 (2011), 69.
- Romain Pr vost, Emily Whiting, Sylvain Lefebvre, and Olga Sorkine-Hornung. 2013. Make It Stand: Balancing Shapes for 3D Fabrication. *ACM Trans. Graph.* 32, 4, Article 81 (July 2013), 10 pages.
- Brian Rooks. 2006. The harmonious robot. *Industrial Robot: An International Journal* 33, 2 (2006), 125–130.
- Nicholas Roy, Paul Newman, and Siddhartha Srinivasa. 2013. *Tendon-Driven Variable Impedance Control Using Reinforcement Learning*. MIT Press, 504–.
- D. C. Rucker and R. J. Webster III. 2011. Statics and Dynamics of Continuum Robots With General Tendon Routing and External Loading. *IEEE Transactions on Robotics* 27, 6 (Dec 2011), 1033–1044. <https://doi.org/10.1109/TRO.2011.2160469>
- Prashant Sachdeva, Shinjiro Sueda, Susanne Bradley, Mikhail Fain, and Dinesh K. Pai. 2015. Biomechanical Simulation and Control of Hands and Tendinous Systems. *ACM Trans. Graph.* 34, 4, Article 42 (July 2015), 10 pages.
- D. Sawada and R. Ozawa. 2012. Joint control of tendon-driven mechanisms with branching tendons. In *2012 IEEE International Conference on Robotics and Automation*. 1501–1507.
- M lina Skouras, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, and Markus Gross. 2013. Computational design of actuated deformable characters. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)* 32, 4 (July 2013), 82:1–82:10.
- A. T. Spr witz, M. Ajalloeian, A. Tuleu, and A. J. Ijspeert. 2014. Kinematic primitives for walking and trotting gaits of a quadruped robot with compliant legs. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)* 8 (2014).
- Shinjiro Sueda, Garrett L. Jones, David I. W. Levin, and Dinesh K. Pai. 2011. Large-scale Dynamic Simulation of Highly Constrained Strands. *ACM Trans. Graph.* 30, 4, Article 39 (July 2011), 10 pages.
- Shinjiro Sueda, Andrew Kaufman, and Dinesh K. Pai. 2008. Musculotendon Simulation for Hand Animation. *ACM Trans. Graph.* 27, 3, Article 83 (Aug. 2008), 8 pages.
- Bernhard Thomaszewski, Stelian Coros, Damien Gauge, Vittorio Megaro, Eitan Grinspun, and Markus Gross. 2014. Computational Design of Linkage-based Characters. *ACM Trans. Graph.* 33, 4, Article 64 (July 2014), 9 pages.
- J. P. Whitney, M. F. Glisson, E. L. Brockmeyer, and J. K. Hodgins. 2014. A low-friction passive fluid transmission and fluid-tendon soft actuator. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2801–2808.
- Andrew Witkin and Michael Kass. 1988. Spacetime Constraints. *SIGGRAPH Comput. Graph.* 22, 4 (June 1988), 159–168. <https://doi.org/10.1145/378456.378507>
- Lifeng Zhu, Weiwei Xu, John Snyder, Yang Liu, Guoping Wang, and Baining Guo. 2012. Motion-guided Mechanical Toy Modeling. *ACM Trans. Graph.* 31, 6, Article 127 (Nov. 2012), 10 pages.