Dynamic Modeling and Motion Planning for Marionettes: Rigid Bodies Articulated by Massless Strings

Elliot Johnson and Todd D. Murphey
Electrical and Computer Engineering
University of Colorado at Boulder
Boulder, Colorado 80309
erjohnso@colorado.edu and murphey@colorado.edu

***Abstract*— We consider the problem of modeling a robotic marionette. Marionettes are highly under-actuated systems that can only be controlled remotely by moving strings. We present a mixed dynamic-kinematic modeling technique that removes the controller dynamics from the marionette, resulting in a clean abstraction that represents the dynamics of the marionette in a natural way. As an example, a model is derived for a single arm moving in a plane. A model for a three-dimensional marionettes is also shown. Finally, an expansive-space tree (EST) motion planner is used to find a path from an input configuration to a goal for a puppet arm with seven degrees of freedom.**

I. INTRODUCTION

Marionettes are puppets that are supported and controlled by manipulating strings tied to points on the marionette. Marionettes have been used for entertainment for centuries. These under-actuated, non-linear, and highly coupled systems can become expressive characters capable of extremely precise and refined movements in the hands of an experienced artist.

In light of this capability, there is interest in developing robotic marionettes. Such systems can be used as research platforms in various areas. There are open research questions in controllers and motion planners for systems that are indirectly actuated. Such problems have been considered for controlling an object hanging from a crane. Additionally, modes of human physical rehabilitation can use pulling on wires to adjust motion of injured people.

There is also ongoing effort to study interaction between humans and robots so that systems can communicate more effectively using expressive body language while restricted by less articulated motor ability. Mapping expressive gestures to marionettes with limited mobility while preserving their expressiveness has been extensively studied by professional puppeteers. A robotic marionette is a useful platform for studying such mapping techniques and generalizing them to more robotics systems.

A prototype puppet stand and its planned successor are shown in Fig. 1. The prototype stand uses six stepper motors to control strings attached to the arms, legs, and head. The length of the strings can be varied, but the strings do not move. Lateral motions for the hands are achieved by attaching two strings to each hand. The stand is controlled using a National Instruments FPGA board with a LabVIEW VI.

An accurate and fast model is an important part of a controller or motion planner for a robotic marionette. In this paper, we present a useful model for the dynamics of a marionette. However, the marionette differs from typical mechanical system in important ways that make modeling the system non-trivial.

Fig. 1. A model of a robotic marionette stand and a photograph of the experimental marionette stand under development.

Mechanical systems are often modeled in two ways. In a dynamic model, all of the forces and inertial bodies of the system are included and used to determine the system's motion within a static environment. A subset of these can be reduced to kinematic models in which the actuators are assumed to be strong enough to track any desired configuration path or that the system intrinsically follows kinematic paths [2].

Neither of these approaches are appropriate for a marionette. A dynamic model must include the both the marionette and the driving mechanisms (so long as the end of the string actuating the marionette is not inertially fixed). The model is therefore tightly coupled to the particular implementation it was derived from. However, there are certainly characteristics of a marionette's dynamics that are independent of the method used to pull its strings. A kinematic model that captures the full range of motion cannot be derived for an under-actuated system such as the marionette.

A better approach is to consider the marionette as two distinct systems. The marionette is considered an unactuated, dynamic system. The string-driving mechanism is a fully actuated system that can be reduced to a kinematic model with velocity inputs. The systems are coupled by the strings, which are introduced to the model as non-holonomic constraints.

Such an arrangement allows us to the study the intrinsic dynamics of the marionette rather than new dynamics introduced by the actuator mechanism. If the dynamics of a particular implementation wish to be studied, the kinematic reduction can easily be replaced with a full dynamic model.

After presenting this method, we discuss simulation results for a full three-dimensional marionette and a three-dimensional marionette arm. An expansive-space tree (EST) motion planner is used to create a motion plan for the arm.

II. PREVIOUS WORK

Physics-based simulation techniques are used extensively in computer graphics and electronic entertainment. A number of fast methods for large physical simulations have been developed [16], [1]. These model systems as sets of rigid bodies joined by constraints. Over time, errors due to numeric integration violate the constraints so spring forces are introduced to restore them. The simulations are realistic, but may not be accurate enough for optimal control applications.

Xing and Chen [17], [6] have developed a robotic marionette that synthesizes motion from a library of behavior primitives. The primitives are designed manually and refined experimentally. A controller combines the primitives to create the requested motions. Very expressive motions are possible, but the primitive library and implementation are specific to the form of the marionette.

Yamane et al [18] present a control system to use motion capture data to drive an automated marionette. This technique uses an inverse kinematic model to determine the appropriate string inputs. The string dynamics are modeled as decoupled swinging dynamics.

The inverse-kinematic method maps motion capture data to the puppet very well, but the controller is limited by the simplified dynamic model. A marionette performance will typically exploit the unique dynamic properties of the system to effectively communicate with the audience. For instance, acrobatic marionettes typically use specialized handles designed to excite particular dynamic modes [8]. If an automated marionette is to be comparable to a traditional performer, the control system will need a model that includes many of the unusual aspects of the system dynamics such as the global coupling and swinging dynamics.

III. TWO DIMENSIONAL MARIONETTE ARM DYNAMICAL MODEL

To illustrate the primary modeling issues associated with a marionette, we initially consider the two-dimensional marionette arm shown in Fig. 2. The arm consists of two rigid links with known masses and moments of inertia. The shoulder is fixed in place and the arm can rotate at the shoulder and elbow. The arm is controlled using a massless string connected to the tip of the arm. The length of the string (L) can be varied and the position of the string (x) can be moved horizontally. The configuration of the system is $q = [\theta_1 \ \theta_2 \ x \ L]^T$.

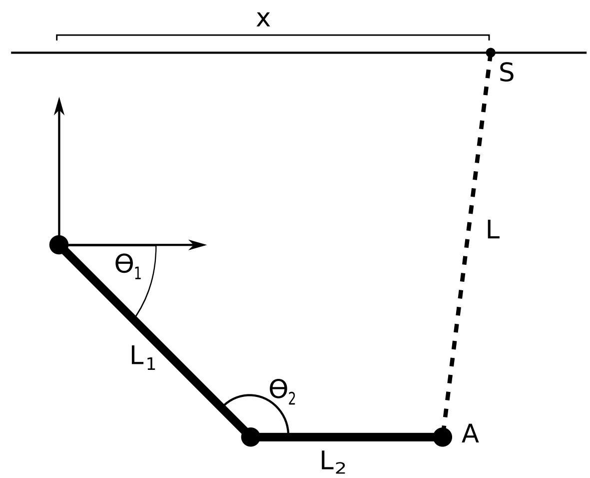

Fig. 2. The 2D Marionette Arm

The arm is modeled using well-known Euler-Lagrange methods for articulated mechanical systems [15]. The system is similar to a closed-chain of rigid bodies, but differs in two important ways. First, one of the linkages has no mass–one therefore cannot consider applying a force to the link or use the inertia of the link. Including the parameters that define the string link (L and x) in the configuration will results in a globally degenerate inertia matrix and, correspondingly, degenerate equations of motion. Associating a mass with the string parameters to avoid this difficulty introduces other problems–a small mass yields stiff differential equations for the motion while a large mass will unrealistically affect the dynamics of the system.

The second difference is that the string represents a prismatic joint because the length of the string can change. This essentially couples the marionette model to the mechanism used to articulate the marionette. To preserve abstraction and simplify the modeling, we desire to separate the marionette dynamics from the actuator dynamics and controller performance.

The solution is to treat the string as a constraint that indirectly applies the system inputs. In this approach, we model the marionette as a dynamical system with a set of non-holonomic string constraints. The system inputs are the velocities of the string lengths and positions and are introduced to the system through Pfaffian constraints. (Note that generally these constraints do not need to always be satisfied. If the z component of the force enforcing the constraint is greater than the force due to gravity, then the constraint will be broken and the string will go slack.) We refer to this approach as the mixed dynamic-kinematic model of the marionette.

A. Euler-Lagrange Model

To derive the equations of motion, the (inertial) base coordinate frame for the system is chosen to be at base of the shoulder. The center of mass for each link is located at the geometric midpoint of the link. For each link, we define an inertia matrix $\mathbb{I}_i \in \mathbb{R}^{3\times3}$

$$\mathbb{I}_i = \begin{bmatrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & \mathcal{I}_i \end{bmatrix} \tag{1}$$

where m_i and $\mathcal{I}_i$ are the mass and rotational inertia of the i-th link, respectively.

The system's Lagrangian can be fully described using θ_1 and θ_2. We define the dynamic configuration, $q_D = [\theta_1\ \theta_2] \in \mathcal{Q}_D$ to distinguish it from the full system configuration.

Let $g_i : \mathcal{Q}_D \rightarrow \mathrm{SE}(2)$ (we use matrix homogeneous representations of $SE(n)$) define the coordinate frame for the center of mass of link i at a given configuration relative to the base frame. For the arm,

$$g_1(q_D) = R(\theta_1)\, T_x\left(\frac{L_1}{2}\right) \tag{2}$$

$$g_2(q_D) = R(\theta_1)\, T_x(L_1)\, R(\theta_2)\, T_x\left(\frac{L_2}{2}\right) \tag{3}$$

where $R(\theta) \in \mathrm{SE}(2)$ performs a rotation by angle θ and $T_x(d) \in \mathrm{SE}(2)$ translates by distance d along the x-axis.

The body velocity $V_i^b : T\mathcal{Q}_D \rightarrow \mathbb{R}^{3\times3}$ of each frame is calculated as:

$$V_i^b(q_D, \dot{q}_D) = g_i^{-1}\frac{dg_i}{dq_D}\dot{q}_D \tag{4}$$

The kinetic and potential energies for each link can now be defined[1]

$$T_i(q_D, \dot{q}_D) = \frac{1}{2}\check{V^b}_i^T \mathbb{I}_i \check{V^b}_i \tag{5}$$

$$V_i(q_D) = [0\ m_i\ 0]\, \check{g}_i \tag{6}$$

The Lagrangian for the system is the sum of the kinetic energy minus the potential energy over all the masses.

$$L(q_D, \dot{q}_D) = \sum_{i=1}^{2}\left(T_i(q_D, \dot{q}_D) - V_i(q_D)\right) \tag{7}$$

The equations of motions are easily derived from the Lagrangian using the Euler-Lagrange equations:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial \dot{q}} = u$$

which are standard equations for modeling mechanical systems [15], [7]. They are expressed in standard form as:

$$M(q_D)\ddot{q}_D + C(q_D, \dot{q}_D)\dot{q}_D + g(q_D) = u \tag{8}$$

where u represents generalized forces on the system, $M(q_D)$ is the inertia matrix, $C(q_D, \dot{q}_D)$ is the Coriolis matrix, and $g(q_D)$ contains gravitational terms. Note that u is not related to the arm's inputs. It can be used to introduce other forces on the arm such as damping. The string forces will be introduced in the following sections.

[1] We use the operator "unhat" such that for $x = \begin{bmatrix} 0 & -\omega & x \\ \omega & 0 & y \\ 0 & 0 & 1 \end{bmatrix}$, $\check{x} = [x\ y\ \omega]^T$

B. Wire Constraint

The dynamic puppet is coupled to the kinematic controller through the string. The string is considered massless and modeled as a constraint. The constraint is analogous to a marble in a bowl where the marble cannot penetrate the bowl, but is free to leave the surface. Initially, we derive the constraint as a rigid wire that strictly controls the distance between two points.

If the wire is not changing length and the point S is not moving (i.e., x and L are not changing), the wire defines a holonomic constraint on the distance between the ends of the wire, P_A and P_S:

$$h(q) = ||P_A(\theta_1, \theta_2) - P_S(x)|| - L = 0$$

That is, it restricts the allowable configurations of the system. When x and L may vary, then we simply rewrite the equation as a Pfaffian constraint

$$A(q)\dot{q} = \frac{\partial h}{\partial q}\dot{q} = 0$$

to be used with the Euler-Lagrange model, where $\dot{x}$ and $\dot{L}$ will become the control inputs to the system.

Finally, the constraint must be rearranged to accommodate the two types of configuration variable. In the previous section, the dynamic configuration q_D was introduced. It is complemented by the kinematic configuration which includes the variables that will be treated kinematically: $q_K = [x\ L]$. Including this distinction is simply a matter of rewriting the constraint as:

$$A(q)\dot{q} = [A_D(q)\ A_K(q)]\begin{bmatrix} \dot{q}_D \\ \dot{q}_K \end{bmatrix} = 0 \tag{9}$$

where $A_D(q) = \frac{\partial h}{\partial q_D}$ and $A_K(q) = \frac{\partial h}{\partial q_K}$.

C. Mixed Dynamic-Kinematic Model

The constraints are included in the dynamic model using Lagrange multipliers, λ, associated with the Lagrange-d'Alhembert principle for constrained mechanical motion:

$$M(q_D)\ddot{q}_D + C(q_D, \dot{q}_D)\dot{q}_D + g(q_D) = u + A_D^T(q_D)\lambda. \tag{10}$$

Equation 10 includes the new term, $A^T(q_D)\lambda$, which is the force applied on the system by the constraints. Equation 9 is differentiated to provide a second equation:

$$A_D\ddot{q}_D + \dot{A}_D\dot{q}_D + A_K\ddot{q}_K + \dot{A}_K\dot{q}_K = 0. \tag{11}$$

Finally, the inputs to the system are the kinematic configuration variables q_K:

$$\ddot{q}_K = u_L \tag{12}$$

Equations 10, 11, and 12 provide a complete set of equations for modeling the marionette for a given set of inputs. We require that the velocity inputs are uniformly continuous so that its derivative is well defined, and that the initial configuration and velocity satisfy the constraint. To integrate the model, Equations 10 and 11 are used to solve for λ and $\ddot{q}_D$. Once $\dot{q}_D$ and $\ddot{q_D}$ are found, the equations can be integrated using any ODE numeric integrator.

The mixed model does not depend on the dynamics of the string actuators in any way, thus providing a clean abstraction between two distinctly different parts of the system. In the case that the inputs are fixed to be constant, the mixed model reduces to the same dynamics that are found using a standard Euler-Lagrange approach.

IV. String Constraints

The wire constraints can be modified to behave like strings, which only enforce a maximum distance between two points, by placing two restrictions on the constraint force. The force must be positive and the can only be applied when the distance between the string endpoints equals the length of the string. We encode these restrictions using δ_1 and δ_2 and calculate the string forces, λ_S, from the original wire forces, λ_W.

$$\delta_1 = \begin{cases} 1 & \lambda_W \geq 0 \\ 0 & \lambda_W < 0 \end{cases}$$

$$\delta_2 = \begin{cases} 1 & ||P_A - P_S|| \geq L \\ 0 & ||P_A - P_S|| < L \end{cases}$$

$$\lambda_S = \delta_1 \delta_2 \lambda_W$$

When the constraint violates either restriction, the constraint force is coerced to zero and the system reduces to Equ. 8. Transitions back to the enabled state require special treatment.

The Pfaffian constraint assumes that initial system velocity does not violate the constraint and applies the necessary force to prevent the constraint from being violated.

The velocity is no longer restricted once the string is disabled. When the constraint is reintroduced to the system, the system velocity must instanteously change to satisfy the constraint. The instantaneously change is accomplished by modeling the transition as a collision involving an impulse.

We set the change in the generalized momentum [13] equal to the collision impulse

$$\Delta p = M(\dot{q}_f - \dot{q}_0) = \vec{K} \tag{13}$$

where $\vec{K}$ is the impulse and $\dot{q}_0$ and $\dot{q}_f$ are the system velocities immediately before and after the collision, respectively. The impulse will be in the same direction as a force applied by a string, so we use the same strategy used with Lagrange multipliers to separate the impulse

$$K = A_D^T \lambda$$

into known and unknown parts.

$$\dot{q}_f = M^{-1} A_D^T \lambda + \dot{q}_0 \tag{14}$$

The collision is considered completely inelastic so that the final velocity immediately satisfies the constraint equation

$$A_D \dot{q}_f + A_K \dot{q}_K = A_D(M^{-1} A_D^T \lambda + \dot{q}_0) + A_K \dot{q}_K = 0$$

rather than bounce upward. We solve for the impulse magnitude

$$\lambda = -(A_D M^{-1} A_D^T)^{-1} (A_D \dot{q}_0 + A_K \dot{q}_K) \tag{15}$$

and use Equ. 14 to determine $\dot{q}_f$. Note that this is equivalent to projecting the velocity using the inertia tensor as the metric–see [7].

When the transition is detected, the integrator is stopped. The appropriate impulse is applied to the arm, the constraint is enabled, and the integrator is restarted.

V. Full Three Dimensional Marionette Dynamic Model

Extending this model to more interesting puppets is straightforward and suitable for automation. In our implementation, a configuration file defines a puppet as a tree of coordinate frames. Masses and strings are attached to different frames. The configuration is used by a `Mathematica` script to symbolically solve for $M(q_D)$, $C(q_D, \dot{q}_D)$, $g(q_D)$, and the constraint matrices. These are exported to a simulator implemented in C.

Fig. 3 shows a complete three dimensional model of a marionette. The dynamic puppet has 25 degrees of freedom (DOF). There are seven strings which can be moved in a plane and wound in and out which introduce 21 kinematic DOF. Though computationally intensive, the model exhibits the rich dynamical behavior of a marionette. Simulation videos of the full three dimensional puppet and other models may be seen at *http://puppeteer.colorado.edu/*.

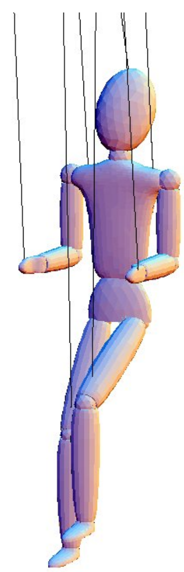

Fig. 3. Three Dimensional Marionette

VI. Validity of Mixed Kinematic-Dynamic Modeling

It is worth noting that the validity of the partial kinematic reduction just used in the mixed dynamic-kinematic model can be verified as follows. In all that follows, we assume that the mechanical system controlling the puppet is fully-actuated (i.e., there is an input for every degree of freedom of the system). The mechanical system including both the puppet and the mechanism controlling the puppet can be described using the constrained Euler-Lagrange equations with $q = [q_D, q_K, q_M]$, where q_D is the configuration of the puppet and q_M is the configuration of the mechanism controlling the puppet. Equivalently, the equations of motion can be written as:

$$\tilde{\nabla}_{\dot{q}}\dot{q} = u_i Y^i$$

where $\tilde{\nabla}$ is the constrained affine connection u^i are the m inputs, and Y_i are the associated input vector fields. (See [2] for a complete description of this formalism) In this context, a system is *kinematically reducible* [3], [2], [4], [5], [14][2] (i.e, all paths on the configuration manifold Q correspond to trajectories on TQ and vice-versa) if $\langle Y_i, Y_j\rangle \in span\{Y_k | 1 \leq k \leq m\}$ where $\langle X, Y\rangle = \tilde{\nabla}_X Y + \tilde{\nabla}_Y X$. Now, because the strings are massless, the inertia tensor is block diagonal in $[q_D, q_K]$. That is, the kinetic energy KE satisfies

$$KE = \frac{1}{2}\frac{d}{dt}([q_D, q_K]) \begin{bmatrix} M_D & 0 \\ 0 & M_K \end{bmatrix} \frac{d}{dt}\left(\begin{bmatrix} q_D \\ q_K \end{bmatrix}\right).$$

This allows us to address the constraints independently so that

$$\begin{aligned} \tilde{\nabla}_{\dot{q}_M}\dot{q}_M &= u_i Y^i \\ \tilde{\nabla}_{\dot{q}_D}\dot{q}_D &= 0. \end{aligned}$$

Because the mechanism controlling the puppet is *assumed* to be fully actuated, the first equation for the mechanism dynamics trivially satisfies the condition $\langle Y_i, Y_j\rangle \in span\{Y_k | 1 \leq k \leq m\}$ for kinematic reducibility (again, see [2] for details of this construction). Hence, separating the kinematic reduction of the mechanism controlling the puppet from the (*not* kinematically reducible) dynamics of the puppet is a mechanically valid description of the system. However, we should emphasize that this is only true when the strings are assumed to be massless.

VII. Motion Planning

As an example application of the model, we have developed a primitive motion planner for three dimensional puppet arm in Fig. 4 based on the Expansive-Space Tree (EST) algorithm [7]. The arm has four dynamic DOF and three kinematic inputs.

The algorithm consists of a collection of configurations, $\mathcal{Q}$, and a desired final configuration, q_f. $\mathcal{Q}$ is initialized with a starting configuration. The motion planner selects a random configuration $q_i \in \mathbb{Q}$ and generates a random input for the system. The model is integrated forward from q_i for a short time using the random input. The final state of the simulation is added to $\mathcal{Q}$.

[2]We do not go into the details of this analysis as it would take us far afield from the basic subject matter. However, the covariant derivative is a fundamental tool in establishing kinematic reducibility, so we introduce it here and point the interested reader to references for further information.

This process is repeated for a large number of iterations. This builds a tree of reachable configurations and the inputs required to move between connected configurations. Eventually, the algorithm generates a configuration near q_f, providing a path from q_0 to q_f.

The performance of the motion planner is drastically improved by biasing the random selection of configurations to favor those closer to the desired configuration. The video attachment for this paper demonstrates a plan generated by this algorithm to move between two configurations while avoid obstacles.

The algorithm is slow and inefficient, but it provides a brute-force way to do motion planning and obstacle avoidance for an under-actuated system like the marionette.

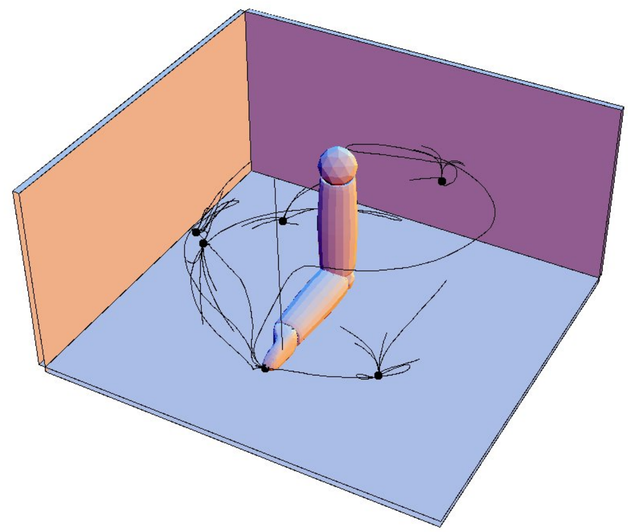

Fig. 4. An expansive-space tree is used to determine a path between two configurations. The black dots show nodes in the search tree and the plot out the path taken to traverse the tree.

VIII. Conclusions and Future Works

The mixed dynamic-kinematic model is a useful tool for modeling marionette dynamics. Using generalized coordinates and Lagrange's equation preserves the mechanical structure of the system and provides a concise expression of the dynamics. The abstraction between the actuator and marionette simplifies the model while maintaining the full behavior of the system.

The purpose of this work is to develop a model suitable for optimal control and motion planning. The current implementation is too slow to do either for a full three-dimensional marionette. The full dynamics must be reduced to a more computationally feasible model. The kinematic-dynamic abstraction is a successful step in this direction.

The string constraint model further simplifies the model. Marionette models developed for computer graphics applications [9] often treat the strings as a chain of small masses joined by stiff springs. These extra components are unnecessary with the string constraint, reducing the number of bodies to simulate and avoiding the numerically undesirable features of stiff springs.

The Euler-Lagrange structure will also allow the model to use variational integrators to improve performance. Variational integrators can allow large simulation time steps while guaranteeing energy conservation during the integration[12], [11], [10].

We wish to develop an automated marionette to explore choreography as an abstract controls concept. In the context of a marionette performance, choreography is essentially an abstract way of defining a trajectory for the system. In this framework, a user can write a play by defining when and where events should occur which would then be compiled by the controller into a complete set of collision-free trajectories for each puppet. Before these concepts can be fully explored, however, several more components of the system will need be developed including a motion planner and model that includes interactions between multiple puppets and the environment.

IX. ACKNOWLEDGMENTS

The authors gratefully acknowledge discussions with Annie Peterle, a puppeteer that has given us much insight into the basic problems of dynamics of string puppets as well as the Atlanta Center for Puppetry Arts for their ongoing contributions to this project.

This material is based upon work partially supported by the National Science Foundation under Grant No. 0546430. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] D. Baraff. Non-penetrating rigid body simulation. In *State of the Art Reports*, 1993.

[2] F. Bullo and A.D. Lewis. *Geometric Control of Mechanical Systems*. Number 49 in Texts in Applied Mathematics. Springer-Verlag, 2004.

[3] F. Bullo and A.D. Lewis. Low-order controllability and kinematic reductions for affine connection control systems. *SIAM Journal on Control and Optimization*, 44(3):885–908, 2005.

[4] F. Bullo, A.D. Lewis, and K.M. Lynch. Controllable kinematic reductions for mechanical systems: Concepts, computational tools, and examples. *Int Symp on Math. Theory of Networks and Systems (MTNS)*, August 2002.

[5] F. Bullo and K. M. Lynch. Kinematic controllability for decoupled trajectory planning in underactuated mechanical systems. *IEEE Transactions on Robotics and Automation*, 17(4):402–412, August 2001.

[6] I.-M. Chen, R. Tay, S.S. Xing, and S.H. Yeo. Marionette: From traditional manipulation to robotic manipulation. In *International Symposium History of Machines and Mechanisms*, Cassiano, Italy, 2004.

[7] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. *Principles of Robot Motion*. The MIT Press, 2005.

[8] P. Fraser. *Puppet Circus*. Plays, Inc., 1971.

[9] S. Kim, X. Zhang, and Y. Kim. Haptic puppetry for interactive games. In *International Conference on E-learning and Games*, 2006.

[10] A. Lew, J. E. Marsden, M. Ortiz, and M. West. Asynchronous variational integrators. *Arch. Rational Mech. Anal.*, 167:85–146, 2003.

[11] A. Lew, J. E. Marsden, M. Ortiz, and M. West. An overview of variational integrators. In *Finite Element Methods: 1970's and Beyond*, pages 98–115, 2004.

[12] A. Lew, J. E. Marsden, M. Ortiz, and M. West. Variational time integrators. *Int. J. Numer. Methods Engrg*, 60:153–212, 2004.

[13] J. E. Marsden and T.S. Ratiu. *Introduction to Mechanics and Symmetry: A Basic Exposition of Classical Mechanical Systems*. Springer, 2002.

[14] T. D. Murphey and J. W. Burdick. The power dissipation method and kinematic reducibility of multiple model robotic systems. *IEEE Transactions on Robotics*, 22(4):694–710, August 2006.

[15] R.M. Murray, Z. Li, and S.S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[16] A. Witkin, M. Gleicher, and W. Welch. Interactive dynamics. In *Computer Graphics*, 1990.

[17] S.S. Xing and I.-M Chen. Design expressive behaviors for robotic puppet. In *7th International Conference on Control, Automation, Robotics, Vision*, pages 378–383, Singapore, 2002.

[18] K. Yamane, J.K. Hodgins, and H.B. Brown. Controlling a marionette with human motion capture data. In *International Conference on Robotics and Automation*, 2003.