

Optimal Robot Motions for Physical Criteria

J. E. Bobrow, B. Martin, G. Sohl, and E. C. Wang

*Dept. of Mechanical and Aerospace Engineering
University of California, Irvine
Irvine, CA 92697*

F. C. Park* and Junggon Kim

*School of Mechanical and Aerospace Engineering
Seoul National University
Seoul 151-742, Korea*

Received 16 August 2001; accepted 16 August 2001

This paper presents an optimization-based framework for emulating the low-level capabilities of human motor coordination and learning. Our approach rests on the observation that in most biological motor learning scenarios some form of optimization with respect to a physical criterion is taking place. By appealing to techniques from the theory of Lie groups, we are able to formulate the equations of motion of complex multibody systems in such a way that the resulting optimization problems can be solved reliably and efficiently—the key lies in the ability to compute exact analytic gradients of the objective function without resorting to numerical approximations. The methodology is illustrated via a wide range of optimized, “natural” motions for robots performing various human-like tasks—for example, power lifting, diving, and gymnastics. © 2001 John Wiley & Sons, Inc.

1. INTRODUCTION

Of all the remarkable physical abilities of humans, motor control is the skill that is most often taken for granted, as it seems to require the least conscious ef-

fort on our part. Only when a particular motor skill is impaired or lost do we then begin to appreciate the difficulty of the overall motor control and learning task. It comes as no surprise that one encounters these exact same difficulties when attempting to program a robot to perform natural, human-like motions; this is the general theme that we address in this paper.

Although we perform everyday motions quite effortlessly, it is all too easy to forget that some of these motor skills are acquired only after considerable

*To whom all correspondence should be addressed; e-mail: fcp@plaza.snu.ac.kr.

Contract grant sponsors: NSF grants INT-9727424, IRI-9711782.

Contract grant sponsors: KOSEF Int. Coll. Res., KIST Service Robot Project.

effort, in some cases after years of practice. From a child learning to throw a ball or swing a bat, to an adult perfecting her golf swing or practicing calligraphy, or a gymnast mastering a new tumbling routine—these seemingly distinct motor skills become a part of the motion vocabulary only after years of practice.

Yet we can still infer some common sequence of events in the motor learning process. Upon first learning the skill, the resulting motion is usually crude and awkward, lacking grace and fluidity—it appears that initially only a few degrees of freedom are deliberately controlled, with the remaining degrees of freedom either locked in place or “pulled along” by mechanical and other built-in constraints of the human body. Then, as the motion is practiced, the once-locked degrees of freedom now begin to unlock, and move in coordination with the controlled primary degrees of freedom. When the motion is perfected, all the degrees of freedom work synchronously to create a fluid and natural-looking motion. The resulting motion is moreover optimized with respect to some physical criterion.

There is another common feature to the motion learning process: the diminishing reliance on feedback as learning takes place. Initially, the motion will be guided and corrected by some form of sensory feedback, for example, visual or tactile. With practice, the resulting motion appears more and more as if a preloaded motor program is being played—in control theory terminology, this transition can be described as moving from a closed-loop to an open-loop control structure as learning occurs.

Clearly motor learning encompasses a much broader range of issues than the ones touched upon above. There are, for example, issues related to force and compliance control, interaction with the environment, integration of visual and other sensory information—and perhaps the most difficult of all to capture, at least in a mathematically precise way—higher-level intelligence in learning. Our aim in this paper is, however, quite specific: to emulate the low-level capabilities of human motor coordination and learning within the framework of optimal control theory. Our approach is based on the simple observation that, in nearly all of the motor learning scenarios that we have observed, some form of optimization with respect to a physical criterion is taking place.

There is ample biological evidence to justify an optimization-based approach to motor control and learning. Indeed, in the literature one can find many optimal control-based studies of various human motions, for example, maximum-height jumping,¹ and voluntary arm movements.² In addition to the more

obvious optimization criteria like minimum energy or control effort, strategies that involve minimizing the derivative of acceleration (or jerk),³ as well as muscle or metabolic energy costs,⁴ have also been examined in the context of specific arm motions. Models for human motor learning and control that take into account both the muscle dynamics and features of the neural system have been proposed in, for example, refs. 5–7; ref. 8 presents an interesting study on the qualitative dynamics of the sit-to-stand movement.

Extensive studies of the cerebellum and its role in motor learning and control have also been performed.^{9,10} The most predominant models adopt a cybernetic metaphor, in which the body is treated as a machine that receives sensory inputs and generates motor outputs. There are proponents of the equilibrium point hypothesis,¹¹ which states that motions are generated according to a potential field determined by the endpoints of the movement. Still others have proposed that the nervous system performs inverse dynamics to generate the motor commands. Although a commonly cited counterargument to this is that such a strategy places too great a computational burden on the nervous system, and that accurate dynamic models are difficult to obtain, recent research also shows that it is possible to identify accurate internal models from movement data, and that such strategies can be successfully implemented in robots (see ref. 12 and the references cited therein). There have also been approaches to motor coordination and learning based on intelligent control,¹³ as well as distributed and hierarchical approaches inspired by biological systems.^{14,15}

From an engineering perspective, an optimization-based approach to motion generation usually comes to mind as the first reasonable thing to try. Past approaches have usually met failure, however, because the complexity of the governing equations of motion usually led to intractable optimization problems. One of the contributions of this paper is that by appealing to techniques from the theory of Lie groups, we can formulate the equations of motion of even complicated multibody systems like the human body in such a way as to render the optimization problem tractable. In many cases the solutions can even be obtained quite efficiently and in a numerically robust way. The key, as we discuss below, lies in the ability to compute exact analytic gradients of the objective function, without resorting to expensive and inaccurate numerical approximations that are often the cause of instability and lack of convergence.

With such motion optimization algorithms at hand, it now becomes possible to realize the notion

of a “motion compiler,” similar in function to a programming language compiler. One view of a motion program is as a concatenation of simpler motion primitives, to which suitable scaling and time warping can be applied. The compiler’s role then is to optimize this sequence of motion primitives with respect to some performance criterion. In this sense the motion compiler can be viewed as a choreographer—it pieces and blends a sequence of crude basic motions into a fluid and artistic dance. In the context of this motion compiler paradigm, we see many potential uses for our optimization tools. For example, motions generated with our optimization algorithms can be used as training data for neural network–based learning schemes, for example. Principal components of the optimized motion can also be extracted and used to create primitives for a richer motion vocabulary.

We begin by briefly describing the dynamic modeling and optimization algorithms developed using techniques from Lie group theory; some of the basic results specific to fully actuated serial chains have been reported in ref. 22, 26. In this paper we extend the class to include tree-structure mechanisms with end-effector constraints, and to underactuated robots. We then illustrate a wide range of optimized, “natural” motions for robots performing various human-like tasks, for example, power lifting, diving, and gymnastics. Although our primary focus will be on generating minimum control effort motions, it should be evident that our algorithms can be straightforwardly generalized to other physical criteria. What is also apparent from these examples is that in many cases the optimized motions are remarkably similar to those performed by their biological counterparts, notwithstanding the results of previous investigations on human arm motions.

2. GEOMETRIC MOTION OPTIMIZATION

For our representation of robotic systems and their dynamics, we use a recently developed set of analytical tools for multibody systems analysis based on the mathematics of Lie groups and Lie algebras.^{24,26} We refer to our approach as a “geometric formulation” as opposed to the “algebraic formulation” used by most robotics researchers. In the traditional algebraic formulation, a rigid motion can be represented with the Denavit-Hartenberg (DH) parameters as a 4×4 homogeneous transformation $T(\theta, d)$, where θ is the rotation about the z -axis and d is the translation along it. For a prismatic joint, d varies while θ is held constant.

For a revolute joint, θ varies while d is held constant. With the geometric formulation, for either type of joint the transformation has the form

$$T(\theta, d) = e^{Ax}M$$

where $x = \theta$ for a revolute joint or $x = d$ for a prismatic joint, A contains the joint axis or direction, and M is a constant ($M = T(0, d)$ for a revolute joint, $M = T(\theta, 0)$ for a prismatic joint). The derivative of the latter form with respect to the joint displacement x is just $dT/dx = Ae^{Ax}M$. If one uses the DH parameters, one must distinguish between joint types and carry the derivatives through to the sine and cosine terms. In the geometric version, one never has to consider the sine and cosine terms. The simplicity and clarity of the geometric version follows through to our computations and makes new developments in dynamics and motion optimization possible.

2.1. Kinematics

Our entire framework for robot design and motion programming is based on an understanding of the geometry of rigid-body motions, or the *Euclidean group*, hereafter denoted $SE(3)$. Its subgroup $SO(3)$ denotes the group of 3×3 proper rotation matrices. Both $SO(3)$ and $SE(3)$ have the structure of both a differentiable manifold and an algebraic group, and are examples of a *Lie group*.

As discussed in the previous section, one of our principal tools in multibody systems analysis is the matrix exponential mapping onto $SE(3)$. Explicit formulas for the map $\exp : se(3) \rightarrow SE(3)$ and its inverse $\log : SE(3) \rightarrow se(3)$ can be derived²⁴; here $se(3)$ denotes the Lie algebra of $SE(3)$. Although $SE(3)$ is not a vector space, $se(3)$ is: the log formula provides a set of *canonical coordinates* for representing neighborhoods of $SE(3)$ as open sets in a vector space.

The use of matrix exponentials to represent the link-to-link transformations for robot systems allows one to clarify the kinematic and dynamic equations. In the case of open chains containing prismatic or revolute joints, the forward kinematics can be written as a product of matrix exponentials.¹⁶ Specifically, given a choice of inertial and tool reference frames, and a zero position for the mechanism, the forward kinematics can be written uniquely as

$$f(q_1, \dots, q_n) = e^{A_1 q_1} \dots e^{A_n q_n}$$

where q_1, \dots, q_n are joint variables, and $A_1, \dots, A_n \in se(3)$. The kinematics of closed chains can be obtained

by further adding a set of algebraic constraints.

The POE formula is an attractive way of systematizing results from classical screw theory, and moreover brings powerful results from differential geometry to bear on the study of mechanisms. Because the POE formula is based on completely standard mathematical definitions and concepts, it eliminates the complex rules and insight that is often necessary in understanding classical screw theory. It has the further advantage of treating prismatic and revolute joints in a uniform way, and does not suffer from numerical ill-conditioning like the Denavit-Hartenberg parameters.

2.2. Dynamics

In order to determine optimal, natural motions for the multibody systems of interest, a complete dynamic model is needed. Within the fixed kinematic topology, our motion compiler can vary the parameters of the model to find the optimal motion, or motor program, for whatever performance measure is selected. The dynamic equations for an open kinematic chain can be written in the form

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (1)$$

where q denotes the joint position vector, $M(q)$ is the inertia matrix, and C and G represent the Coriolis/centrifugal and gravity terms, respectively. In Park, Bobrow, and Ploen,²⁶ a Lie group formulation of the dynamics has been developed, in which closed-form expressions for the inertia matrix and Coriolis terms are available that are particularly amenable to differentiation. Using our representation, the forward and inverse dynamics can also be computed efficiently with $O(n)$ recursive algorithms. The inverse dynamics algorithm is shown in Figure 1. In this algorithm, $V_i \in \mathfrak{se}(3)$ is the linear and angular velocity of link i , \mathcal{W} is the applied force and moment, J is a 6×6 matrix of mass and inertia, S_i is the joint screw, and Ad and ad are standard operators from differential geometry.²⁴ A useful computational feature of this algorithm is that no distinction must be made for revolute or prismatic joints.

2.3. Motion Optimization

We assume that the generation of the motor program involves the minimization of a cost function of the form

$$\text{Minimize}_{\tau(t)} \quad J = \Phi(q, \dot{q}, t_f) + \int_0^{t_f} L(q, \dot{q}, \tau, t) dt$$

• Initialization

$$V_0 = \dot{V}_0 = \mathcal{W}_{n+1} = 0$$

• Forward recursion: for $i = 1$ to n do

$$\begin{aligned} T_{i-1,i} &= M_i e^{S_i \dot{q}_i} \\ V_i &= \text{Ad}_{T_{i-1,i}}(V_{i-1}) + S_i \dot{q}_i \\ \dot{V}_i &= S_i \dot{\dot{q}}_i + \text{Ad}_{T_{i-1,i}}(\dot{V}_{i-1}) \\ &\quad + \left[\text{Ad}_{T_{i-1,i}}(V_{i-1}), S_i \dot{q}_i \right] \end{aligned}$$

• Backward recursion: for $i = n$ to 1 do

$$\begin{aligned} \mathcal{W}_i &= \text{Ad}_{T_{i+1,i}}^*(\mathcal{W}_{i+1}) + J_i \dot{V}_i - \text{ad}_{V_i}^*(J_i V_i) \\ \tau_i &= S_i^T \mathcal{W}_i \end{aligned}$$

Figure 1. The POE recursive Newton-Euler inverse dynamics algorithm.

$$\text{subject to} \quad (1)$$

$$\text{and} \quad q(0) = q_0, \dot{q}(0) = 0 \quad (3)$$

$$q(t_f) = q_f, \dot{q}(t_f) = 0, \quad (4)$$

where, for some of our examples, Φ penalizes deviations from the desired final condition. For most of our examples, the effort $L = \frac{1}{2} \|\tau^e\|^2$ captures the desire to minimize the joint torques. The final time t_f may be either free or fixed in our formulation.

A local solution to the preceding optimal control problem is found by assuming that the joint coordinates $q(t)$ in (1) are parameterized by B-splines, and varying these parameters in the following manner. The B-spline curve depends on the blending, or basis, functions $B_i(t)$, and the control points $P = \{p_1, \dots, p_m\}$, with $p_i \in \mathfrak{R}^6$. The joint trajectories then have the form $q = q(t, P)$ with

$$q(t, P) = \sum_{i=1}^m B_i(t) p_i \quad (5)$$

An example of a trajectory and the basis functions is shown in Figure 2. The control points p_i of the spline have only a local effect on the curve geometry (see Chapter 6 in ref. 17), so given any t there will a maximum of four nonzero $B_i(t)$ in (5) for a cubic spline. In addition, the convex hull property of B-splines

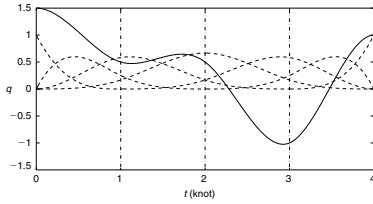


Figure 2. An example trajectory generated as the sum of cubic B-spline basis functions. The dotted lines are the cubic basis functions.

makes them useful for smoothing or approximating data. The fact that $\sum_{i=1}^m B_i(t) = 1$ also gives the desirable property that limits on joint displacements can be imposed through limits on the spline parameters p_i . That is, if one constrains $p_i \leq \bar{q}$, then it follows that $q \leq \bar{q}$.

We should note that the use of B-spline polynomials as the basic primitives upon which all of our motions are developed is consistent with recent results in neuroscience. In ref. 23, it was observed clinically that when human subjects move their hand in a circular motion, the trajectory obtained can be best described as a summation of “bell shaped” basis functions. These functions are then translated and scaled to find the best match to the human movement. We are achieving the same basic effect through (5).

The parameter optimization equivalent of the original optimal control problem is

$$\text{Minimize}_P \quad J(P) = \Phi(P, t_f) + \int_0^{t_f} L(P, t) dt \quad (6)$$

$$\text{subject to} \quad \underline{q} \leq p_i \leq \bar{q}, \quad i = 1 \dots m \quad (7)$$

With this approach, $\tau = \tau(P, t); q, \dot{q}$, and \ddot{q} all are given functions of t and P from (5) and its time-derivatives, and hence τ is an explicit function of the spline parameters through (1). By the proper choice of the spline basis functions at both ends of the joint trajectory, the path-end conditions (3) and (4) can be satisfied.

We have converted the original problem into a parameter optimization problem with no nonlinear constraints, and efficient quasi-Newton algorithms can then be used to solve the problem. However, for assured convergence of these algorithms, two conditions must be met: the second derivatives of $J(P)$ must be bounded, and every approximate Hessian (found,

for example, from a BFGS update²⁵) used in the quasi-Newton algorithm must remain positive definite with bounded condition number.¹⁸

Due to the complexity of the dynamic equations of motion, most previous solutions to nontrivial optimal control problems for robotic systems use finite difference gradient approximations of $J(P)$. In these cases, it is usually not possible to ensure a bounded condition number of the approximate Hessian, and the algorithms usually terminate prematurely. In order to compute the gradient of the cost functional, we note that

$$\nabla_P J = \int_0^{t_f} \tau^T \cdot (\nabla_P \tau) dt \quad (8)$$

The most significant step for this gradient is computing the derivatives of the joint torques with respect to the path parameters P . We compute these derivatives analytically by differentiating the recursive dynamics shown in Figure 1. The resulting recursive algorithm for the gradient is shown in Figure 3.

• Initialization

$$\frac{dV_0}{dp_i} = \frac{d\dot{V}_0}{dp_i} = \frac{dW_{n+1}}{dp_i} = 0, \quad \forall p_i \in P$$

• Forward recursion: for $i = 1$ to n do

$$\begin{aligned} \frac{dV_i}{dp_i} &= \frac{dq_i}{dp_i} \text{ad}_{\text{Ad}_{T_{i-1,j}}^{(V_{i-1})}} S_i + \text{Ad}_{T_{i-1,j}} \frac{dV_{i-1}}{dp_i} + S_i \frac{d\dot{q}}{dp_i} \\ \frac{d\dot{V}_i}{dp_i} &= \frac{dq_i}{dp_i} \text{ad}_{\text{Ad}_{T_{i-1,j}}^{(\dot{V}_{i-1})}} S_i + \text{Ad}_{T_{i-1,j}} \frac{d\dot{V}_{i-1}}{dp_i} \\ &\quad + \text{ad}_{\frac{dV_i}{dp_i}} S_i \dot{q}_i + \text{ad}_{V_i} S_i \frac{d\dot{q}_i}{dp_i} + S_i \frac{d\ddot{q}}{dp_i} \end{aligned}$$

• Backward recursion: for $i = n$ to 1 do

$$\begin{aligned} \frac{dW_i}{dp_i} &= J_i \frac{d\dot{V}_i}{dp_i} + \text{Ad}_{T_{i+1}}^* \frac{dW_{i+1}}{dp_i} \\ &\quad - \text{ad}_{\frac{dV_i}{dp_i}}^* J_i V_i - \text{ad}_{V_i}^* J_i \frac{dV_i}{dp_i} \\ &\quad - \text{ad}_{\text{Ad}_{S_{i+1}, S_{i+1}} \frac{d\dot{q}_{i+1}}{dp_i}} \text{Ad}_{T_{i+1}}^* W_{i+1} \\ \frac{d\tau_i}{dp_i} &= S_i^T \frac{dW_i}{dp_i} \end{aligned}$$

Figure 3. The POE recursive derivative Newton-Euler inverse dynamics algorithm.

3. SOME SAMPLE MOTIONS

3.1. Planar Weightlifter

Our algorithms have been used to generate optimized motions for several problems (see refs. 19, 21, and 22 for more details). As a first example, we examine a 5R planar model of a weightlifter using realistic link lengths and mass properties, as shown in Figure 4. The initial, improper, lifting technique is shown in the Figure 4(a), with the optimized lift in Figure 4(b). All joints had equal weighting in the cost functional, and the back joint produces most of the torque load in the original lift. This load was taken up more by the leg joints in the optimized lift. Figure 4(c) shows that the integral of the cost function was reduced by an order of magnitude from the original. The major savings in effort comes from reducing the torque applied at the beginning of the motion by the back, and the torque applied in the middle of the motion by the shoulders. Of course, the lifter also passes the weights through its knees, which points out the need for adding barrier avoidance to the optimization problem. The path for this problem was a 5-knot, uniform cubic B-spline. The initial and final costs $J(P)$ for the weightlifter were 2.25×10^7 and 1.44×10^6 respectively. The optimization took approximately 120 seconds to compute on an SGI Indigo 2 running at 150 Mhz.

In the sequence of plots in Figures 4(d-f), the weightlifter problem was solved again with the initial angles free and an added Cartesian constraint that keeps the hands touching the weights at the beginning of the motion. This solves the kinematic redundancy problem in a manner that is optimal for the motion requirement. Figure 4(d) looks similar to the Olympic lift known as "the snatch"—the only difference is that in the snatch the lifter pauses with the weight over his head before lifting with his legs. Figures 4(e) and (f) are local minimizers with different final times; the latter motion has an extra oscillation in which energy is pumped into the system.

3.2. Cooperating Robots—The Rising Camel

As a second example, consider the system of two cooperating robot arms lifting a workpiece or two fingers grasping a common object, shown in Figure 5. The task is to lift the 55 kg object from the initial position shown in the left frame to the final position shown in the right frame. The initial path is specified from intuition as a straight line, as shown in Figure 5(a). The final optimized path is shown in Figure 5(b). There is more than an order of magnitude reduction for the second sequence as compared to the first, with $J = 5.5 \times 10^7$ for the first motion and $J = 1.2 \times 10^6$ for the second motion. It is interesting that the final motion uses the

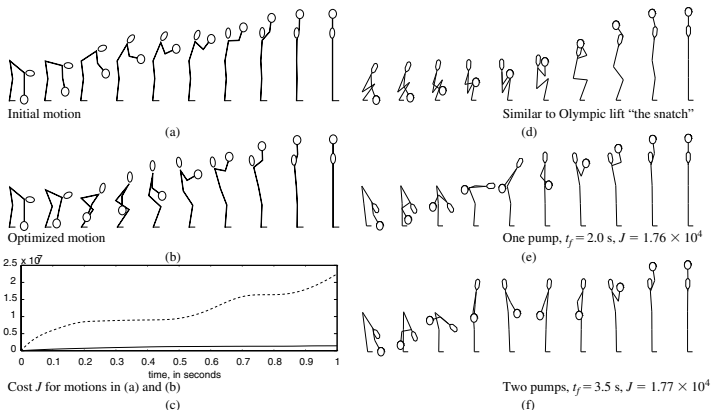


Figure 4. A model weightlifter. (a) is the initial trajectory. (d-f) are local minima found for the free end time problem with free initial joint configurations.

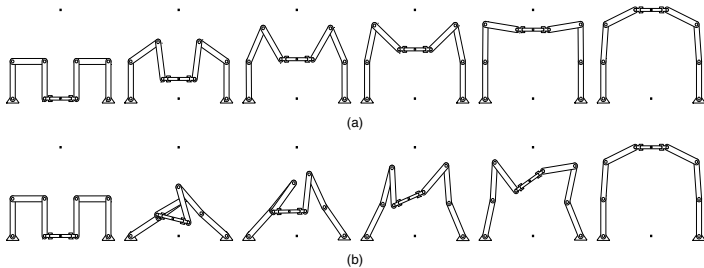


Figure 5. Cooperating robot problem, comparison of (a) original and (b) final motions.

singularities in the closed chain for leverage, rather than trying to avoid them as is common in robotics research. The optimized path looks remarkably similar to the motion that the torso of a large animal such as an elephant or camel would make to stand up.

3.3. Robot Power Lift

Application of the algorithm to a real robot requires accurate modeling of the manipulator dynamics, friction, joint velocity limits, and actuator torque bounds. In ref. 19, the quantity $-wp_w$, where p_w is the payload and w is a constant weighting factor, was added outside the integral of the cost function in (6). This term expressed our desire to maximize the payload lifted by a Puma 762 robot. We added hard constraints into the optimization to handle the velocity limits, and soft constraints to handle the torque limits. The

manufacturer's specifications state that a maximum load of 20 kg can be lifted at a distance of 25 cm from the center of its wrist. The payload is modeled as an ideal plate. The trajectories of the Puma's six joints were parameterized by uniform quintic B-splines.

Three different cases (of one dof, three dofs, and six dofs) are discussed, where

- one dof: the fourth joint is the only joint with freedom to move;
- three dof: the last three joints (wrist) are set free while the others are fixed; and
- six dof: all six joints are free.

In each case, the optimization procedure is initialized with $p_w = 20$ kg, $t_f = 20$ sec, and a prescribed path that moves the fourth joint from the initial configuration to the final one, shown in Figure 6, on a smooth trajectory without any swing and any movement of the other joints. The solutions to the optimization problem for the three cases are shown in Table 1 and Figures 7–9. The resulting paths are motions that routinely swing through singular configurations. These motions were successfully implemented on our Puma 762 robot. Using the six-dof trajectory, we effectively *tripled* the payload limit specified by the manufacturer.

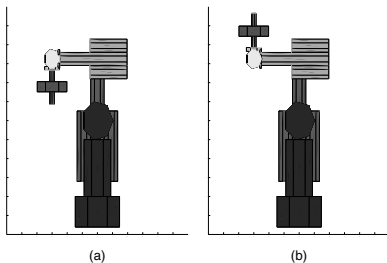


Figure 6. (a) Initial and (b) final configurations of the Puma.

Table 1. The optimal results for three cases.

Case	Initial J_c	Final J_c	Final time	Payload
one dof	-24.2	-42.5	7.8 sec	29.4 kg
three dof	-4.15	-21.1	3.1 sec	46.0 kg
six dof	-181.0	-600.1	11.9 sec	63.2 kg

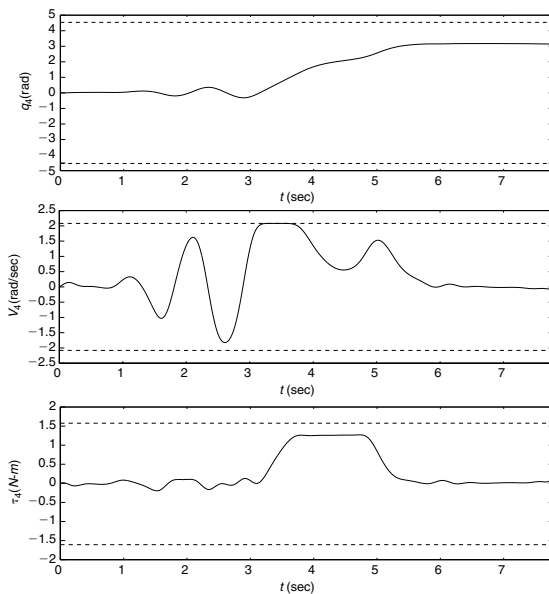


Figure 7. The motion, speed and torque of the fourth joint in the one-dof case (limits: dotted lines).

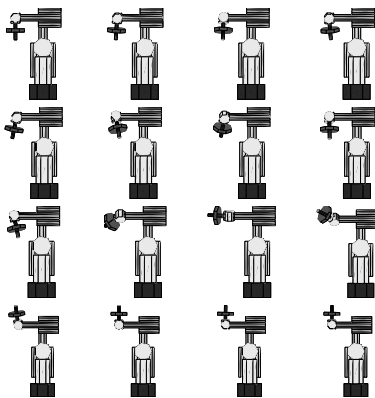


Figure 8. The motion of the wrist in the three-dof case.

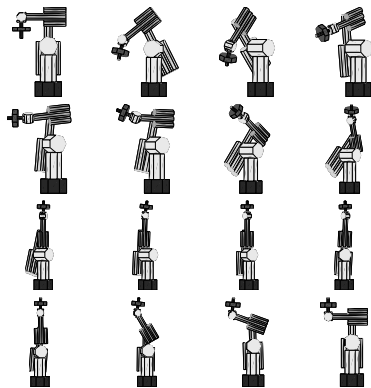


Figure 9. The motion for the six-dof case.

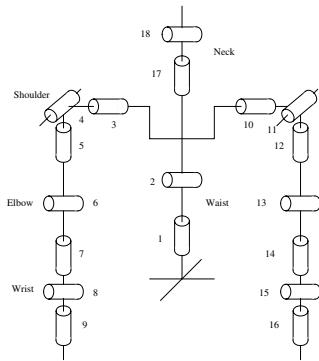


Figure 10. The structure of the humanoid in KIST.

Another application of robot power lifting that we investigated involves a humanoid robot recently developed at the Korea Institute of Science and Technology.²⁰ The robot, shown in Figure 10, consists of two seven-dof arms, a two-dof waist, and a two-dof neck. Since the structure of the robot is tree type, the dynamics and its derivative, shown in Figures 1 and 3, cannot be used directly in this structure. However we can easily extend them to the tree-type structure by replacing the subscripts $i-1$ and $i+1$ with i and μ_i , which represent the precedent and descendant links, respectively. The task is to lift the 5 kg dumbbell with the right arm, and the resulting motion is shown in Figure 11. Altering the constraints imposed on the maximum joint torques, joint limits, and so on, will alter the motion slightly, but overall it seems that the optimized path very closely resembles the motion of human arms.

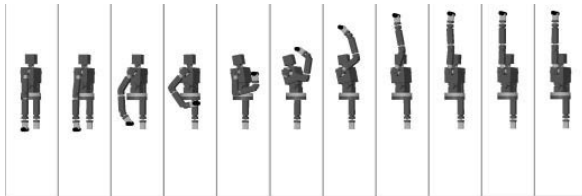


Figure 11. Snapshot of the dumbbell-lifting humanoid.

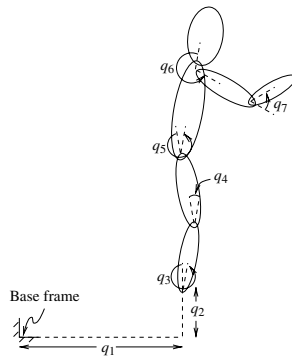


Figure 12. Planar diver model.

3.4. Underactuated Robots—The Diver

The same basic strategy used in the preceding examples can be applied to systems with passive degrees of freedom. However, the dynamics become more complicated since the motion of the passive joints cannot be prescribed directly with B-splines. In ref. 21, the extension to underactuated robots is developed. Consider the case of a planar model of a high diver, as shown in Figure 12. This is similar to the diver analyzed by Crawford and Sastry.¹⁵ We have connected the diver to a fixed base position using three passive joints (two prismatic and one revolute joint). These passive joints exert no force on the diver, allowing the diver to perform as a free flying body. In order to make the diver perform a forward $1\frac{1}{2}$ somersault, we chose a cost function of the form

$$J = c_1(q_3(t_f) - 3\pi)^2 + c_2 \int_0^{t_f} \|\tau\|^2 \quad (9)$$

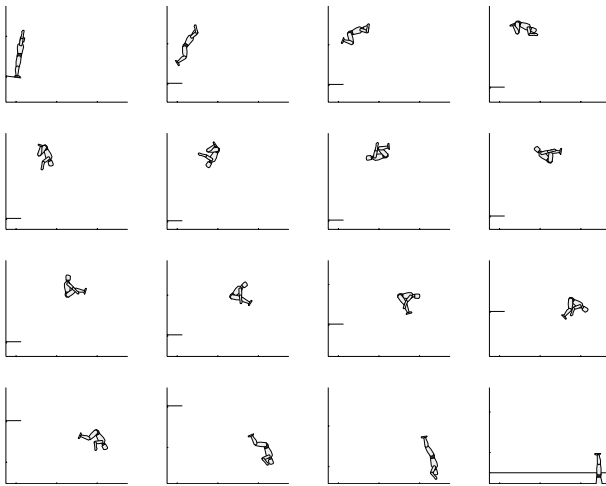


Figure 13. Forward $1\frac{1}{2}$ somersault.

where the first term penalizes any deviation from the desired entry angle and the second term penalizes the effort required to complete the motion. The resulting dive is shown in Figure 13. The human-like features of this dive again demonstrate the potential of our algorithms for use in more complex motion planners.

4. CONCLUSIONS

In this paper, we have presented a methodology for generating robot motions that are optimal with respect to physical criteria. By appealing to techniques from the theory of Lie groups, efficient optimization algorithms can be obtained for a wide variety of mechanisms, including open and closed chains, overactuated and underactuated systems, and so on. Aside from their independent use, these algorithms can form the basis for instilling in robots the ability to emulate the low-level capabilities of human motor coordination and learning, that is, a "motion compiler." In this context, some potential applications for our optimization algorithms include using the optimized motions as training data for, neural network-based learning schemes, for example. While our algorithm produces

reliable and physically plausible solutions, the computational requirements are still significant; using the optimized motions as training data for more advanced learning schemes appears to be a promising approach to generating motions in a more computationally efficient way.

Although the primary focus in our examples has been on minimum control effort motions, our optimization algorithms can, with modest effort, be straightforwardly modified to handle other physical criteria. In this aspect, they can serve as a useful simulation tool for researchers studying biological motor control—by including, for example, the muscle dynamics in the description of the equations of motion, various hypotheses regarding biological motions can be tested via simulation.

REFERENCES

1. M.G. Pandy, F.E. Zajac, E. Sim, and W.S. Levine, An optimal control model for maximum-height human jumping, *J Biomech* 23:(12) (1990), 1185–1198.
2. N. Lan, Analysis of an optimal control model of multi-joint arm movements, *Biol Cybern* 76 (1997), 107–117.

3. G.J. Garvin, M. Zefran, E.A. Henis, and V. Kumar, Two-arm trajectory planning in a manipulation task, *Biol Cybern* 76 (1997), 53–62.
4. R.M. Alexander, A minimum energy cost hypothesis for human arm trajectories, *Biol Cybern* 76 (1997), 97–105.
5. S. Stroeve, Learning combined feedback and feedforward control of a musculoskeletal system, *Biol Cybern* 75 (1996), 73–83.
6. A. Karniel and G.F. Inbar, A model for learning human reaching movements, *Biol Cybern* 77 (1997), 173–185.
7. G.L. Gottlieb, D.M. Corcos, and G.C. Agarwal, Strategies for the control of voluntary movements with one mechanical degree of freedom, *Behavioral Brain Sci* 12 (1989), 189–250.
8. P.G. Roberts and G. McCollum, Dynamics of the sit-to-stand movement, *Biol Cybern* 74 (1996), 147–157.
9. J.C. Houk, J.T. Buckingham, and A.G. Barto, Models of the cerebellum and motor learning, <http://depts.physio.nwu.edu/Physiology/houk/bbs.html>, 1998.
10. E.J. Parkin, Cerebellum and cerebrum in adaptive control and cognition: A review, *Biol Cybern* 77 (1997), 79–87.
11. D.B. Asatryan and A.G. Feldman, Functional tuning of the nervous system with control of movement or maintenance of a steady posture, *Biofizika* 2 (1965), 925–935.
12. C.G. Atkeson, Learning arm kinematics and dynamics, *Ann Rev Neurosci* 12 (1989), 157–183.
13. G. Saridis, Intelligent robotic control, *IEEE Trans Autom Contr* 28(5) (1983), 547–557.
14. F.A. Mussa-Ivaldi, Nonlinear force fields: A distributed system of control primitives for representing and learning movements, *Proc IEEE Int Symp Computational Intelligence in Robotics and Automation*, 1997, pp. 84–90.
15. L.S. Crawford and S.S. Sastry, Biological motor control approaches for a planar diver, *Proc 34th IEEE Conf Dec Contr*, New Orleans, LA, 4, 1995, pp. 3881–3886.
16. R.W. Brockett, Robotic manipulators and the product of exponentials formula, *Proc Int Symp Math Theory Networks Syst*, Beer Sheva, Israel, 1983.
17. I.D. Faux, and M.J. Pratt, *Computational geometry for design and manufacture*, 4th ed., Wiley, New York, 1979.
18. P.E. Gill, W. Murray, and M.H. Wright, *Practical optimization*, Academic Press, 1981.
19. C.E. Wang, W.K. Timoszyk, and J.E. Bobrow, Weighlifting Motion Planning for a Puma 762 Robot, *IEEE Conf on Robotics and Automation*, Detroit, Michigan, April 1999.
20. Y.K. Hwang et al., Human interface, automatic planning, and control of a humanoid robot, *Int J Robot Res* 17(11), 1131–1149.
21. G.A. Sohl and J.E. Bobrow, Generation of optimal motions for underactuated robots, *IEEE Conf on Robotics and Automation*, Detroit, Michigan, April 1999.
22. B.J. Martin and J.E. Bobrow, Minimum effort motions for open chain manipulators with task-dependent end-effector constraints, *Int J Robot Res*, 8(2) (1999), 213–224.
23. H.I. Krebs, N. Hogan, M.L. Aisen, and B.T. Volpe, Robot-aided neurorehabilitation, *IEEE Trans Rehab Eng* 6(1) (1998), 75–87.
24. R.W. Murray, Z. Li, and S.S. Sastry, A mathematical introduction to robotic manipulation, CRC Press, Boca Raton, Florida, 1993.
25. D.G. Luenberger, *Linear and nonlinear programming*, 2nd ed., Addison-Wesley, Reading, MA, 1989.
26. F.C. Park, J.E. Bobrow, and S.R. Ploen, A Lie group formulation of robot dynamics, *Int J Robot Res* 14(6) (1995), 609–618.