Learning Control for Complex Skills

Lara S. Crawford

University of California, Berkeley

http://robotics.eecs.berkeley.edu/~lara/

Controlling Complex Skills

Systems with behavioral goals such as walking or diving present several control challenges:

- Inherently complex dynamics
- Many degrees of freedom
- Desired trajectory unknown; trajectory subordinate to overall behavioral goal

Solutions?

Learning, biologically-inspired architectures

Outline

- The diver problem
  - Kinematic analysis
  - Initial control approaches
- The solution
  - Biological motor control
  - Psychological learning theory
  - Reinforcement learning
- The learning controller
  - Design
  - Results
- Conclusions and future directions

The Diver Problem

Control goal: Execute dive maneuver and enter the water in a fully-extended, vertical position.

A dive is specified by a number of somersaults and twists plus configuration information. Some examples:

jackknife ($\frac{1}{2}$ somersault pike)
$1\frac{1}{2}$ somersault pike
$1\frac{1}{2}$ somersault pike with full twist

throw pike layout

Features of the problem:

- Nonholonomic constraint, nonlinear drift
- Fixed time in which controls can act
- Behavioral goal

Related Research

Nonholonomic systems:

- Physics of orientational maneuvers

  Frolich, 1979; Montgomery, 1990

- Steering for systems without drift, classes of systems with drift

  Walsh and Sastry, 1995; Brockett, 1972; Di Giamberardino, Monaco, and Normand-Cyrot, 1996; Godhavn, Balluchi, Crawford, and Sastry, 1998

State machine approaches:

- Robots and computer animation

  Hodgins and Raibert, 1990; Wooten and Hodgins, 1996

Learning:

- Learning through demonstration

  Schaal, 1997

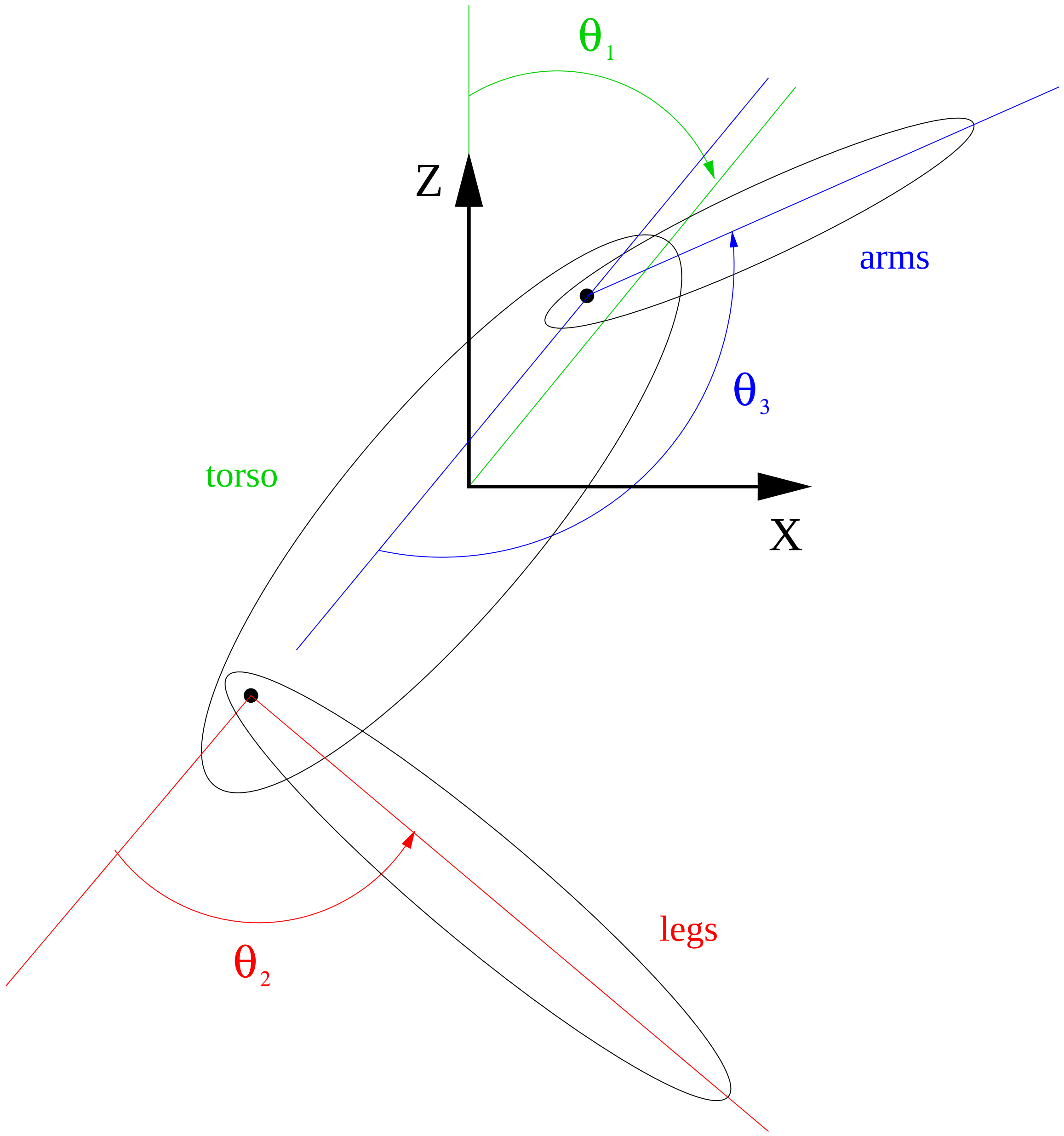
- Control parametrization approaches

  Gorinevsky, Kapitanovsky, and Goldenberg, 1996

- Neuro-dynamic programming

  Bertsekas and Tsitsiklis, 1996, 1995

Planar Diver Model



Kinematic Analysis

Lagrangian:

$$
\begin{aligned}
L &= \Big(\frac{\alpha_1}{2} + \beta\cos\theta_2 + \gamma\sin\theta_2 + \delta\cos\theta_3 + \epsilon\sin\theta_3 \\
&\quad +\zeta\cos(\theta_3-\theta_2) + \eta\sin(\theta_3-\theta_2)\Big)\dot\theta_1^2 \\
&+ \Big(\alpha_2 + \beta\cos\theta_2 + \gamma\sin\theta_2 \\
&\quad +\zeta\cos(\theta_3-\theta_2) + \eta\sin(\theta_3-\theta_2)\Big)\dot\theta_1\dot\theta_2 \\
&+ \Big(\alpha_3 + \delta\cos\theta_3 + \epsilon\sin\theta_3 \\
&\quad +\zeta\cos(\theta_3-\theta_2) + \eta\sin(\theta_3-\theta_2)\Big)\dot\theta_1\dot\theta_3 \\
&+ \Big(\zeta\cos(\theta_3-\theta_2) + \eta\sin(\theta_3-\theta_2)\Big)\dot\theta_2\dot\theta_3 \\
&+ \frac{\alpha_2}{2}\dot\theta_2^2 + \frac{\alpha_3}{2}\dot\theta_3^2
\end{aligned}
$$

Kinematic Analysis

Angular momentum conservation:

$$\frac{\partial L}{\partial \theta_1} = 0 = \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_1}$$

$$\begin{aligned}
\mu \;\; = \;\; & [\alpha_1 + 2\beta \cos\theta_2 + 2\gamma \sin\theta_2 \\
& +2\delta \cos\theta_3 + 2\epsilon \sin\theta_3 \\
& +2\zeta \cos(\theta_3 - \theta_2) + 2\eta \sin(\theta_3 - \theta_2)]\dot{\theta}_1 \\
+ \;\; & [\alpha_2 + \beta \cos\theta_2 + \gamma \sin\theta_2 \\
& +\zeta \cos(\theta_3 - \theta_2) + \eta \sin(\theta_3 - \theta_2)]\dot{\theta}_2 \\
+ \;\; & [\alpha_3 + \delta \cos\theta_3 + \epsilon \sin\theta_3 \\
& +\zeta \cos(\theta_3 - \theta_2 + \eta \sin(\theta_3 - \theta_2)]\dot{\theta}_3
\end{aligned}$$

$$=: \begin{bmatrix} b_1(\theta_2, \theta_3) & b_2(\theta_2, \theta_3) & b_3(\theta_2, \theta_3) \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

This nonintegrable constraint on the joint velocities is called a nonholonomic constraint.

$$\begin{aligned}
\alpha_1 &= I_t + I_l + I_a - \frac{2m_l m_a}{M}(j_{lx}j_{ax} + j_{lz}j_{az}) \\
&\quad + \frac{m_l(M - m_l)}{M}(l_x^2 + l_z^2 + j_{lx}^2 + j_{lz}^2) \\
&\quad + \frac{m_l(M - m_a)}{M}(a_x^2 + a_z^2 + j_{ax}^2 + j_{az}^2) \\
\alpha_2 &= I_l + \frac{m_l(M - m_l)}{M}(l_x^2 + l_z^2) \\
\alpha_3 &= I_a + \frac{m_a(M - m_a)}{M}(a_x^2 + a_z^2) \\
\beta &= \frac{m_a m_l}{M}(-j_{az}l_z - j_{ax}l_x) + \frac{m_l^2}{M}(-j_{lx}l_x - j_{lz}l_z) \\
&\quad + m_l(j_{lx}l_x + j_{lz}l_z) \\
\gamma &= \frac{m_a m_l}{M}(j_{ax}l_z - j_{ax}l_x) + \frac{m_l^2}{M}(l_{lx}l_z - j_{lz}l_x) \\
&\quad - m_l(j_{lx}l_z - j_{lz}l_x) \\
\delta &= \frac{m_a m_l}{M}(-j_{lz}a_z - j_{lx}a_x) + \frac{m_a^2}{M}(-j_{ax}a_x - j_{az}a_z) \\
&\quad + m_a(j_{ax}a_x + j_{az}a_z) \\
\epsilon &= \frac{m_a m_l}{M}(j_{lx}a_z - j_{lz}a_x) + \frac{m_a^2}{M}(j_{ax}a_z - j_{az}a_x) \\
&\quad - m_a(j_{ax}a_z - j_{az}a_x) \\
\zeta &= \frac{m_a m_l}{M}(-a_x l_x - a_z l_z) \\
\eta &= \frac{m_a m_l}{M}(a_z l_x - a_x l_z)
\end{aligned}$$

Control System

Using velocity controls at shoulders and hips:

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \begin{bmatrix} -\frac{b_2}{b_1} \\ 1 \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} -\frac{b_3}{b_1} \\ 0 \\ 1 \end{bmatrix} u_2 + \begin{bmatrix} \frac{\mu}{b_1} \\ 0 \\ 0 \end{bmatrix}$$

$$=: g_1(\theta_2, \theta_3) u_1 + g_2(\theta_2, \theta_3) u_2 + f(\theta_2, \theta_3)$$

Controllability rank condition: If the Lie algebra generated by $\{g_1, g_2\}$ has full rank at a point, the system is locally controllable at that point.

For the diver, we have:

$$[g_1, g_2] = \begin{bmatrix} \frac{1}{b_1^2}\left(b_3 \frac{\partial b_1}{\partial \theta_2} - b_1 \frac{\partial b_3}{\partial \theta_2} - b_2 \frac{\partial b_1}{\partial \theta_3} + b_1 \frac{\partial b_2}{\partial \theta_3}\right) \\ 0 \\ 0 \end{bmatrix}$$

$\{g_1, g_2, [g_1, g_2]\}$ spans the space (except at isolated points) so the system is locally controllable.

Exterior Differential Systems View

$$I = \{\alpha\},$$

$$\alpha = b_1 d\theta_1 + b_2 d\theta_2 + b_3 d\theta_3 - \mu dt$$

$$\begin{aligned}
d\alpha &= db_1 \wedge d\theta_1 + db_2 \wedge d\theta_2 + db_3 \wedge d\theta_3 \\
&= -\frac{\partial b_1}{\partial \theta_2} d\theta_1 \wedge d\theta_2 - \frac{\partial b_1}{\partial \theta_3} d\theta_1 \wedge d\theta_3 \\
&\qquad + \left(\frac{\partial b_3}{\partial \theta_2} - \frac{\partial b_2}{\partial \theta_3} \right) d\theta_2 \wedge d\theta_3 \\
&=: h_1(\theta_2, \theta_3) d\theta_1 \wedge d\theta_2 + h_2(\theta_2, \theta_3) d\theta_1 \wedge d\theta_3 \\
&\qquad + h_3(\theta_2, \theta_3) d\theta_2 \wedge d\theta_3
\end{aligned}$$

$$\begin{aligned}
d\alpha \wedge \alpha &= (h_1 b_3 - h_2 b_2 + h_3 b_1)\ d\theta_1 \wedge d\theta_2 \wedge d\theta_3 \\
&\qquad - h_1 \mu \quad d\theta_1 \wedge d\theta_2 \wedge dt \\
&\qquad - h_2 \mu \quad d\theta_1 \wedge d\theta_3 \wedge dt \\
&\qquad - h_3 \mu \quad d\theta_2 \wedge d\theta_3 \wedge dt
\end{aligned}$$

$$(d\alpha)^2 \wedge \alpha = 0$$

Derived flag:

$$\begin{aligned}
I^{(0)} &= \{\alpha\} \\
I^{(1)} &= \{0\}
\end{aligned}$$

$$
\begin{aligned}
d\alpha &= db_1 \wedge d\theta_1 + db_2 \wedge d\theta_2 + db_3 \wedge d\theta_3 \\
&= -\frac{\partial b_1}{\partial \theta_2} d\theta_1 \wedge d\theta_2 - \frac{\partial b_1}{\partial \theta_3} d\theta_1 \wedge d\theta_3 \\
&\qquad\qquad + \left(\frac{\partial b_3}{\partial \theta_2} - \frac{\partial b_2}{\partial \theta_3} \right) d\theta_2 \wedge d\theta_3 \\
&= -2(\gamma \cos\theta_2 - \beta \sin\theta_2 - \eta \cos(\theta_3 - \theta_2) \\
&\qquad\qquad + \zeta \sin(\theta_3 - \theta_2))\ d\theta_1 \wedge d\theta_2 \\
&\quad -2(\epsilon \cos\theta_3 - \delta \sin\theta_3 + \eta \cos(\theta_3 - \theta_2) \\
&\qquad\qquad - \zeta \sin(\theta_3 - \theta_2))\ d\theta_1 \wedge d\theta_3 \\
&\quad +2(-\eta \cos(\theta_3 - \theta_2) + \zeta \sin(\theta_3 - \theta_2))\ d\theta_2 \wedge d\theta_3
\end{aligned}
$$

$$
\begin{aligned}
d\alpha \wedge \alpha &= [2b_1(-\eta \cos(\theta_3 - \theta_2) + \zeta \sin(\theta_3 - \theta_2)) \\
&\qquad + 2b_2(\epsilon \cos\theta_3 - \delta \sin\theta_3 + \eta \cos(\theta_3 - \theta_2) \\
&\qquad - \zeta \sin(\theta_3 - \theta_2)) - 2b_3(\gamma \cos\theta_2 - \beta \sin\theta_2 \\
&\qquad - \eta \cos(\theta_3 - \theta_2) + \zeta \sin(\theta_3 - \theta_2))] \\
&\qquad\qquad\qquad d\theta_1 \wedge d\theta_2 \wedge d\theta_3 \\
&\quad + 2\mu(\gamma \cos\theta_2 - \beta \sin\theta_2 - \eta \cos(\theta_3 - \theta_2) \\
&\qquad + \zeta \sin(\theta_3 - \theta_2))\ \ d\theta_1 \wedge d\theta_2 \wedge dt \\
&\quad + 2\mu(\epsilon \cos\theta_3 - \delta \sin\theta_3 + \eta \cos(\theta_3 - \theta_2) \\
&\qquad - \zeta \sin(\theta_3 - \theta_2))\ \ d\theta_1 \wedge d\theta_3 \wedge dt \\
&\quad - 2\mu(-\eta \cos(\theta_3 - \theta_2) + \zeta \sin(\theta_3 - \theta_2)) \\
&\qquad\qquad\qquad d\theta_2 \wedge d\theta_3 \wedge dt
\end{aligned}
$$

$$
(d\alpha)^2 \wedge \alpha = 0
$$

Pfaff's theorem: there exist local coordinates (z_1, z_2, z_3) such that

$$\{\alpha\} = \{dz_3 - z_2 dz_1\}$$

Need to find q_1, q_2 which satisfy:

$$\begin{aligned} d\alpha \wedge \alpha \wedge dq_1 &= 0 \\ \alpha \wedge dq_1 &\neq 0 \\ \alpha \wedge dq_1 \wedge dq_2 &= 0 \\ dq_1 \wedge dq_2 &\neq 0 \end{aligned}$$

Then I can be written as

$$\{\alpha\} = \{dq_2 - s dq_1\}$$

and the control system becomes:

$$\begin{aligned} \dot{q}_1 &= u_1 \\ \dot{s} &= u_2 \\ \dot{q} &= s\dot{q}_1 = su_1 \end{aligned}$$

Diver system: simple choices for q_1, q_2 fail. $q_1 = t$ not a solution $\Rightarrow$ system is **not** feedback linearizable.

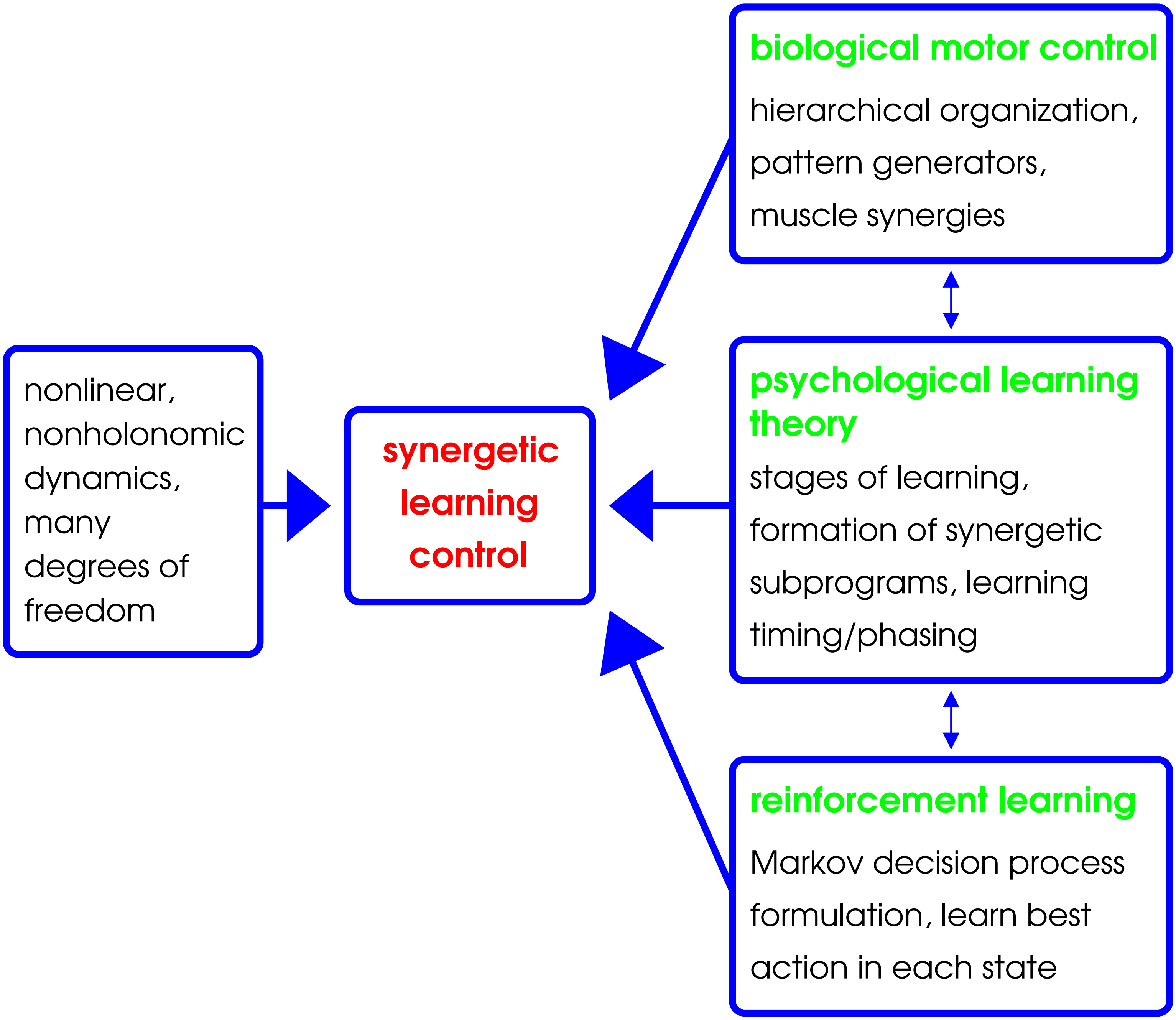
Control Approaches

- Sinusoidal steering with chained form
- Optimal control
- Finite discretizability of nilpotentizable systems (Di Giamberardino, *et. al.*, 1996)
- Bang-bang controls for Lagrangian systems with cyclic coordinates (Godhavn, Balluchi, Crawford, and Sastry, 1998)
- Simple learning approach using parametrized biologically plausible control velocity profiles (Crawford and Sastry, 1995)
- Dynamic learning controller

Outline

- The diver problem
  - Kinematic analysis
  - Initial control approaches
- The solution
  - Biological motor control
  - Psychological learning theory
  - Reinforcement learning
- The learning controller
  - Design
  - Results
- Conclusions and future directions

The Solution



Behavioral/Synergetic Architectures

In a behavioral or synergetic control structure, controls are restricted to a parametrized family, resulting in stereotyped system behavior.

Synergetic organization simplifies the control and learning task by:

- Reducing the number of possible controls
- Representing the controls in a compact, intuitive way
- Allowing generalization from known to new tasks

Examples of behavioral organization in biological motor control hierarchy:

- Pattern generators
  - Rhythmic movements
  - Fast, goal-directed, single-joint movements

- Muscle synergies

- Multi-joint synergies

- Subprograms (parallel or series) in complex skills
  - Walking
  - Slaloming
  - Diving

$\longrightarrow$ hierarchical, behavioral control design

Pattern Generators

Neural pattern generators produce stereotypical control patterns that can be tuned by descending controls.

- Rhythmic movements: walking, running, swimming, breathing, chewing

- Fast, goal-directed, single-joint movements: stereotypical two- or three-pulse EMG profile, one-hump velocity profile

Gottlieb, Corcos, and Agarwal (1989), Gottlieb (1993), PWM/PHM low-level control scheme for fast, single-joint movements

Phases of Learning

- Cognitive stage: Task or skill is analyzed, perhaps put into words. Stage during which the input of a teacher can be effective.

  "Learning by watching"

- Associative (fixation) stage: Performer practices the skill, improving and growing more consistent until errors are rare.

- Autonomous stage: Performer continues practicing until the skill becomes automatic, and requires very little conscious thought.

  "Learning by doing"

Reinforcement Learning

Learning algorithms for problems in a Markov decision process framework:

states $\mathbf{s} \in \mathcal{S}$

actions $\mathbf{v} \in \mathcal{V}(\mathbf{s})$

transition probabilities $p(\mathbf{s_i}, \mathbf{v}, \mathbf{s_j}) = p_{i,j}(\mathbf{v})$

transition costs $c(\mathbf{s_i}, \mathbf{v}, \mathbf{s_j}) = c_{i,j}(\mathbf{v})$

Algorithms learn the minimum expected cost-to-go or value of each state and thus can determine the best policy (choice of action in each state)

Flavors of Reinforcement Learning

Value iteration: Discrete states, table lookup, full model available. Learn $J^*(\mathbf{s})$.

TD(λ): Simulation-based incremental policy evaluation. Learn $J^\mu(\mathbf{s})$, then update policy μ.

Q-learning: No model of transition probabilities and costs; simulation-based. Learn $Q^*(\mathbf{s}, \mathbf{v})$.

Neuro-dynamic programming: Large or continuous state space. Learn parametrized approximations $\tilde{J}(\mathbf{s}, \mathbf{w})$ or $\tilde{Q}(\mathbf{s}, \mathbf{v}, \mathbf{w})$.
Convergence guaranteed only in some cases (TD(λ) with linear approximator).
(Bertsekas & Tsitsiklis)

Diver problem:

No model of transitions $\rightarrow$ Q-learning
Continuous variables $\rightarrow$
function approximator

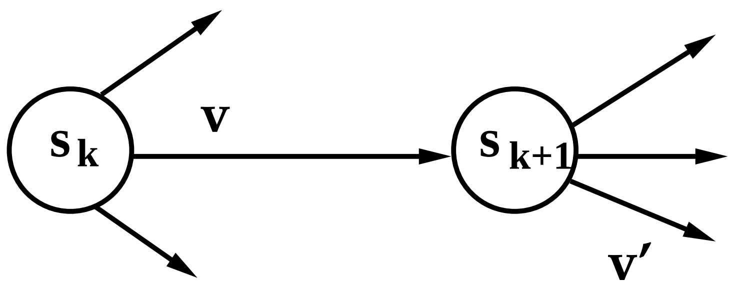
Q-Learning

Define the Q function, $Q(\mathbf{s_k}, \mathbf{v})$, the expected cost-to-go when action $\mathbf{v}$ is chosen in state $\mathbf{s_k}$ and optimal actions are chosen thereafter.

The optimal Q function, $Q^*(\mathbf{s_k}, \mathbf{v})$, is a fixed point of the Bellman equation:

$$Q^*(\mathbf{s_k}, \mathbf{v}) = E\left[c_{k,k+1}(\mathbf{v}) + \min_{\mathbf{v}'} Q^*(\mathbf{s_{k+1}}, \mathbf{v}')\right]$$

With a function approximator, $\tilde{Q} = \tilde{Q}(\mathbf{s_k}, \mathbf{v}, \mathbf{w})$, where $\mathbf{w}$ is a vector of weights.



Q-Learning Algorithm

1. In $\mathbf{s_k}$, choose $\mathbf{v}$ (usually the greedy action) and simulate the state transition.

2. Calculate the temporal difference d_k:

$$d_k \;=\; c_{k,k+1}(\mathbf{v}) + \min_{\mathbf{v}'} Q(\mathbf{s_{k+1}}, \mathbf{v}') - Q(\mathbf{s_k}, \mathbf{v})$$

3. Update Q:

$$Q^{k+1}(\mathbf{s_k}, \mathbf{v}) = Q^k(\mathbf{s_k}, \mathbf{v}) + \alpha_k\, d_k$$

or

$$\mathbf{w}^{k+1} = \mathbf{w}^k + \alpha_k \frac{\partial \tilde{Q}}{\partial \mathbf{w}}(\mathbf{s_k}, \mathbf{v}, \mathbf{w}^k)\, d_k$$

4. Repeat for the new state $\mathbf{s_{k+1}}$.

Outline

- The diver problem
  - Kinematic analysis
  - Initial control approaches
- The solution
  - Biological motor control
  - Psychological learning theory
  - Reinforcement learning
- The learning controller
  - Design
  - Results
- Conclusions and future directions

Controller Design

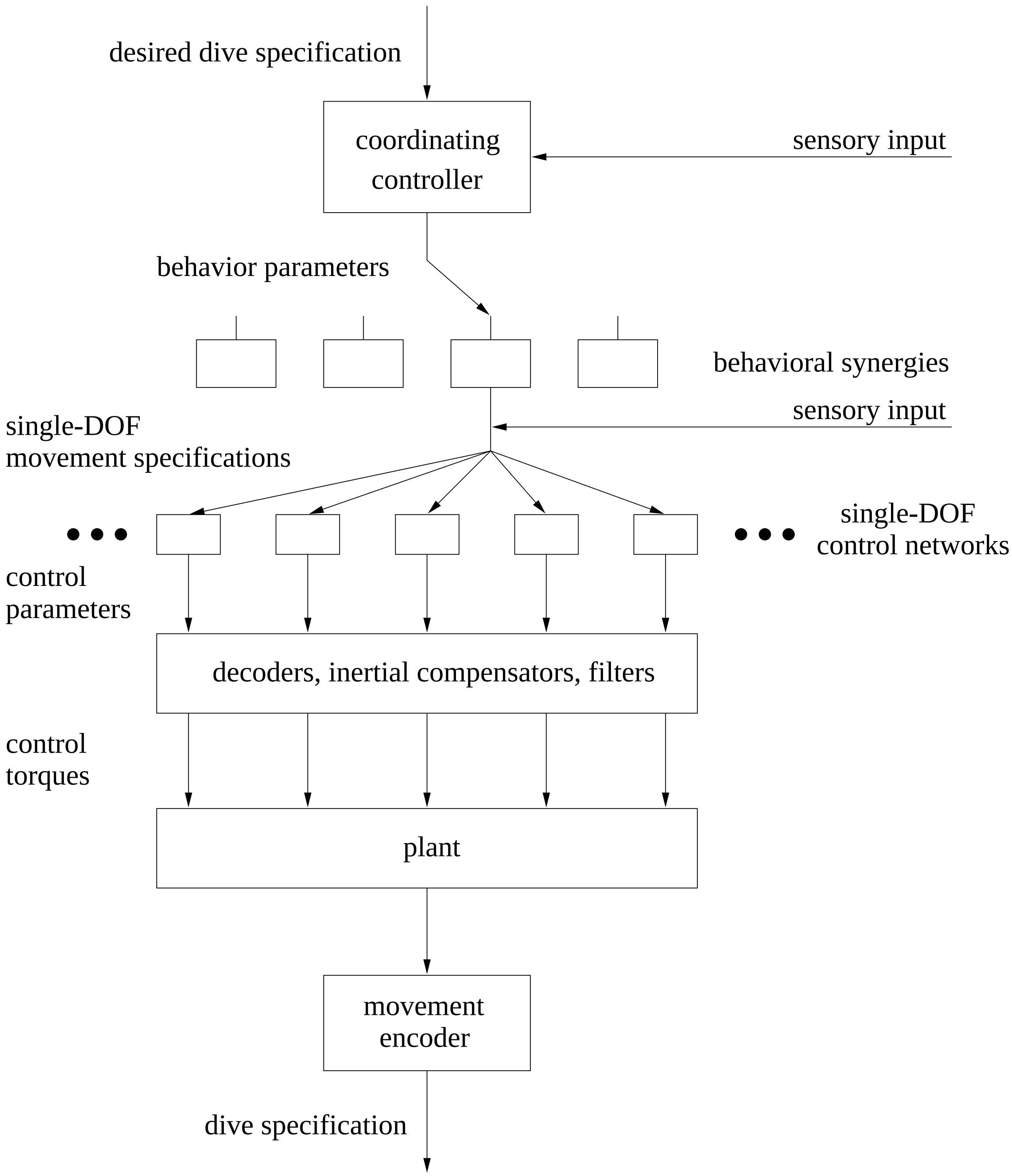
- Hierarchical: Single-DOF controllers, coordinating controller

- Hybrid: Encoders and decoders mediate between regimes

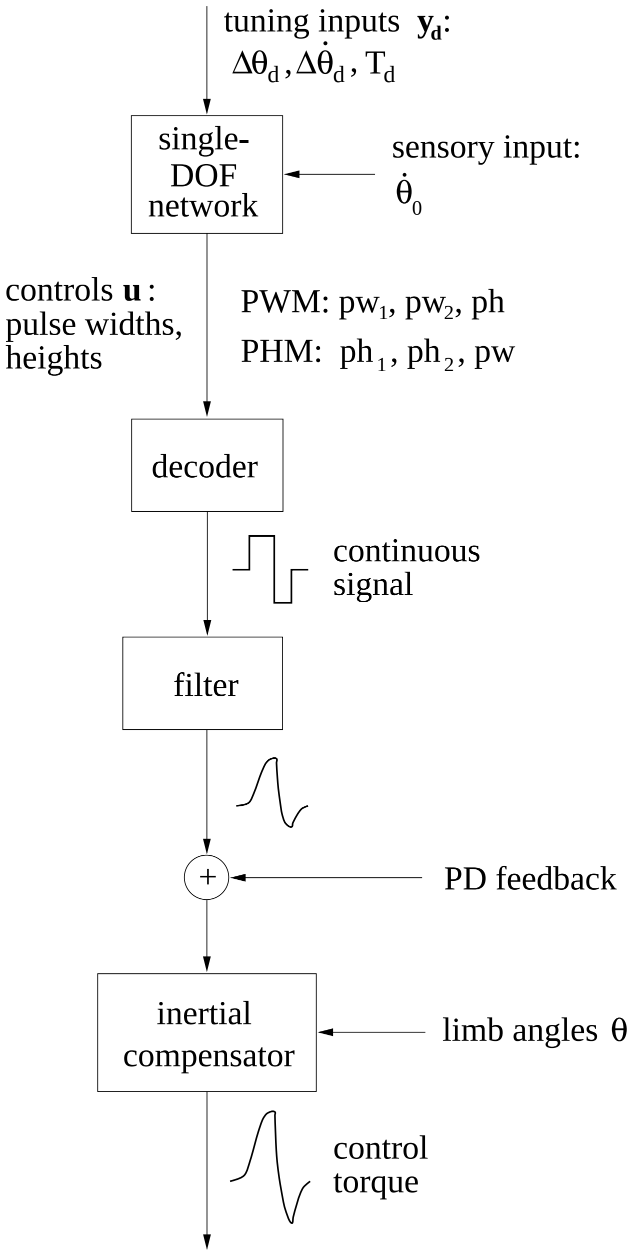
- Biologically-inspired: Synergetic architecture (pattern generators, behavioral subprograms), learning stages

- Learning: Internal models, supervised learning, reinforcement learning

Controller Architecture

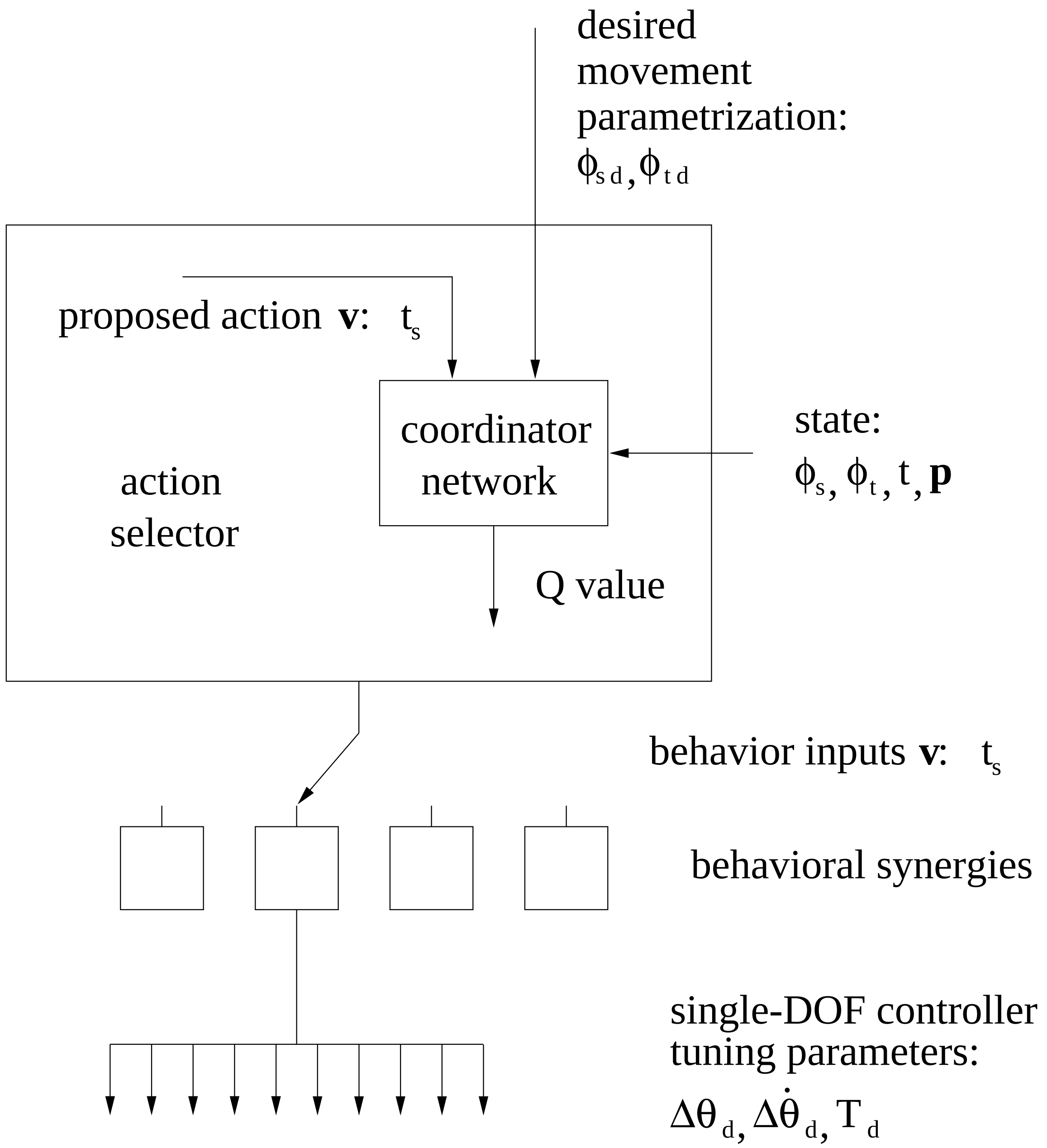


Single-DOF Controller



- Plays role of pattern generator
- PWM / PHM control scheme
- Supervised learning

Coordinating Controller



- Sequential activation of synergies
- Learns timing between movements

Coordinating Controller Learning

- Learning by watching:

  Behavior sequence predefined:

  jackknife = (pike, layout)

  $1\frac{1}{2}$ somersault pike = (pike, layout)

  $1\frac{1}{2}$ somersault twist = (throw, pike, layout)

  Data from Olympic divers used to initialize network weights

- Learning by doing:

  Q-learning with function approximation (neural network)

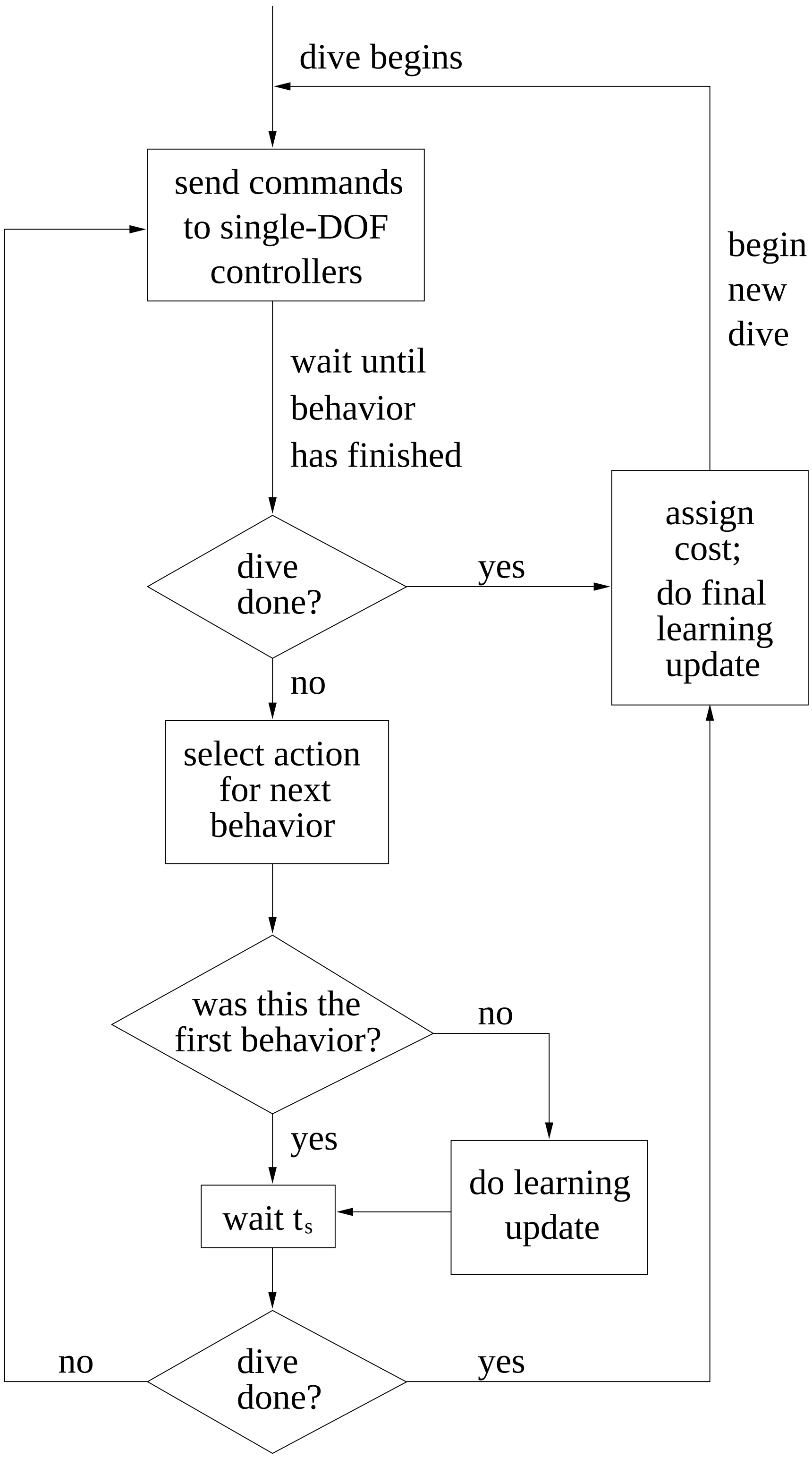
  $\mathbf{s} = [\phi_{sd}, \phi_{td}, \phi_s, \phi_t, t, \mathbf{p}]$

  $\mathbf{v} = [t_s]$

  Cost assessed at end of dive

  Learning by watching initialization promotes convergence

Coordinating Controller Learning



Results

Videotape

Conclusions

The learning controller uses synergetic structure to achieve the simple, parametric representation of controls which makes learning possible.

This approach is useful for complex skills or other problems for which there are:

- A known goal
- Complex or unknown dynamics or system behavior
- Known behavioral/synergetic strategies (cognitive, learning-by-watching information is available)

Future Directions and Applications

- Generalized behaviors
- Continuous time learning methods
- Role of feedback in complex skill learning
- Encoding linguistic movement descriptions into behaviors

- Robotics
- Animation
- Visual learning and gesture recognition
- Complex motor skill training (athletics)
- Rehabilitation

http://robotics.eecs.berkeley.edu/~lara/

Simple Learning Controller

Controls restricted to a family of biologically plausible velocity profiles:

$$
\begin{aligned}
u_1 &= \frac{A}{\sqrt{2\pi\sigma^2}}\left[e^{-\frac{(t-\tau_1)^2}{2\sigma^2}} - e^{-\frac{(t-\tau_2)^2}{2\sigma^2}}\right] \\
u_2 &= -\frac{A}{\sqrt{2\pi\sigma^2}}\left[e^{-\frac{(t-\tau_1)^2}{2\sigma^2}} - e^{-\frac{(t-\tau_2)^2}{2\sigma^2}}\right]
\end{aligned}
$$

Controller tuned by changing parameter vector:

$$
\mathbf{p} = \begin{bmatrix} A \\ \sigma^2 \\ \tau_1 \\ \tau_2 - \tau_1 \end{bmatrix}
$$

Error criterion:

$$
\begin{aligned}
E(\mathbf{p}) = (\text{total rotation} - 540^\circ)^2 + \text{final leg angle}^2 \\
+ \text{final arm angle}^2 + k(\tau_2 + \sigma)^2
\end{aligned}
$$

Simple Learning Algorithm

1. Random iterations: keep best parameter vector and error as starting point for gradient descent

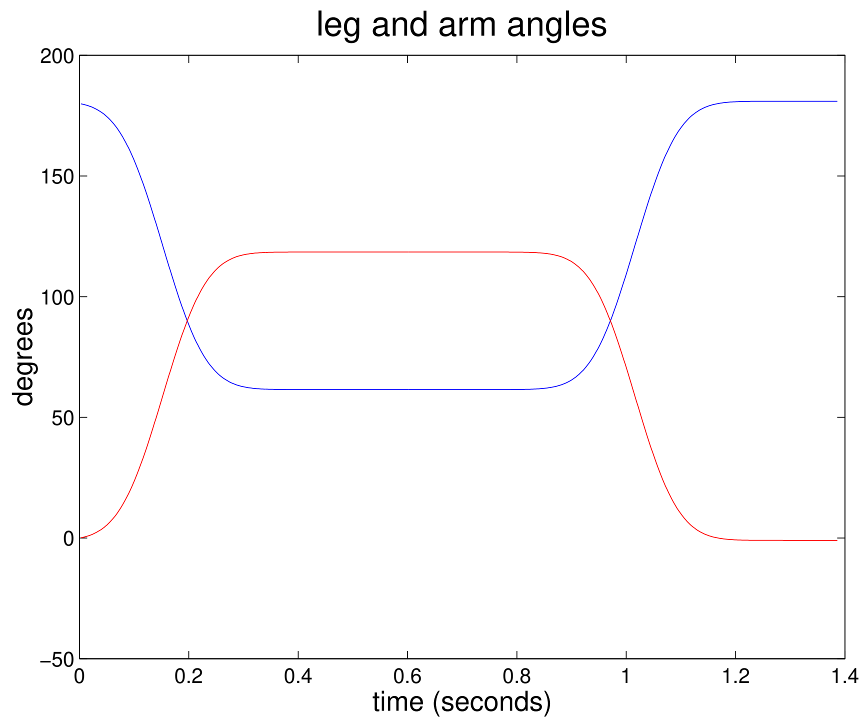
2. a. Update local gradient estimate (decaying average of $\frac{\Delta E}{\Delta \mathbf{p}}$)

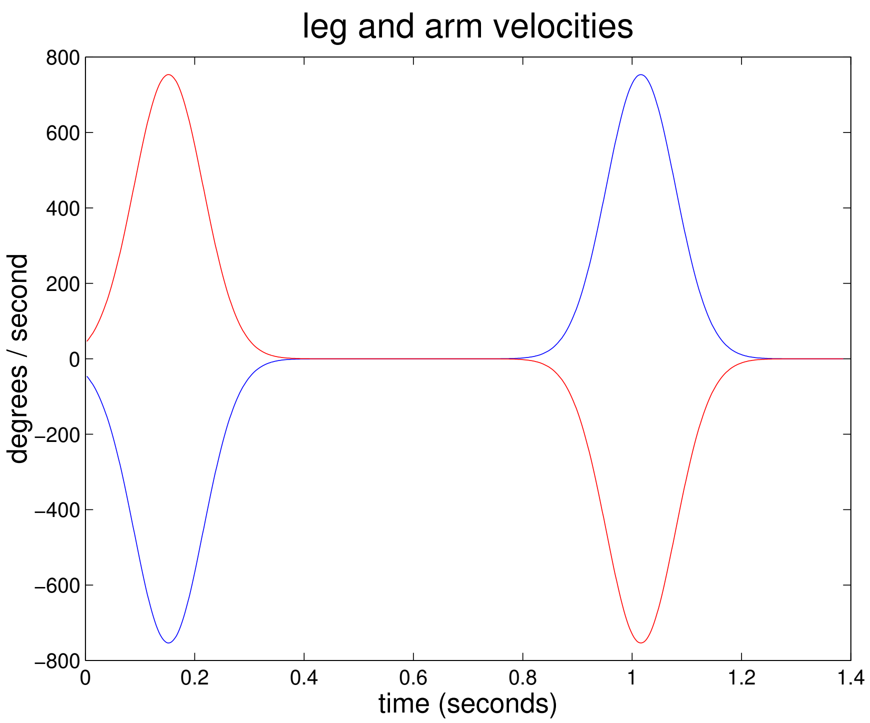
   b. Update parameter vector by gradient descent

   c. Is the new error lower than the old error?
   If so, keep new parameter vector, error

   d. Repeat from 2a until error is acceptable

Simple Learning Algorithm–Results





Simple Learning Algorithm–Results