UMassAmherst

Control Basis: Landscapes of Attractors

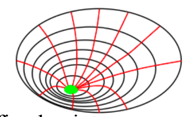
Rod Grupen,
CS 603 Robotics
UMass Amherst

UMassAmherst

Potential Functions

q_2

"funneling"
classifiers in error phase space



$\phi|^{\sigma}_{\tau}$

fixed point sensory geometry
relative to features
afforded by the environment

q_1

UMassAmherst

Control Basis*: Track primitive

objectives x *sensors* x *effectors*

T: TRACK
$a = \phi|^{\sigma}_{\tau}$

action: closed-loop feature (σ) tracker where sensor viewpoint is controlled with kinematic chain τ

state: $\gamma(a)$ = 0 "unknown"
= 1 no reference
= 2 transient
= 3 converged

visual foveation – contact force tracking
any feature of any signal

UMassAmherst

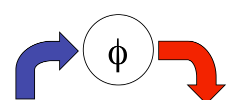
Representational Foundations - the *control basis*

a control theoretic framework with built in *intentions*

multiple
navigation functions

declarative

continuous
sensory signals f_σ



continuous
u_τ motor variables

← | σ | | τ | →

procedural

discrete
sensorimotor
resources

typing - characteristic input and output types
supports multiple implementations

Manfred Huber

UMassAmherst

Control Basis Actions

$$\phi \left| \begin{array}{l} \sigma \longleftarrow \\ \tau \longrightarrow \end{array} \right. \text{ENV}$$

scalar

more rows than columns

$$J = \frac{\partial \phi(\sigma)}{\partial u_\tau} \in R^{1xn}$$

redundant (underconstrained)

vector of changes in setpoints

UMassAmherst

Control Basis Actions

$$\Delta\phi(\sigma) = J \Delta u_\tau$$

$J^\#$ Moore-Penrose pseudoinverse [Appendix A.9]

$$\Delta u_\tau = \kappa[J^\# \Delta\phi(\sigma)]$$

$$J^\# = J^T[JJ^T]^{-1}$$

$$= -\kappa J^\# \phi(\sigma)$$

UMassAmherst

Multi-Objective Actions

$$c2 \triangleleft c1 \Rightarrow \kappa_1 J_1^\# \Delta\phi_1(\sigma_1) + [I - J_1^\# J_1][\kappa_2 J_2^\# \Delta\phi_2(\sigma_2)]$$

$J^\#$ and $(I - J^\# J)$ are orthogonal?

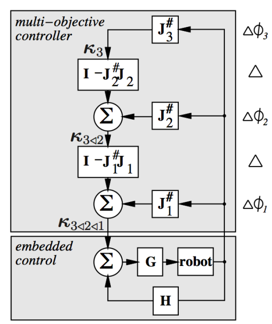
$$(J^\#)^T (I - J^\# J) = 0$$

$$[(I - J^\# J)^T \; J^\#]^T = (J^\# - J^\# J J^\#)^T = 0$$

from the Moore-Penrose conditions

UMassAmherst

Multi-Objective Controllers



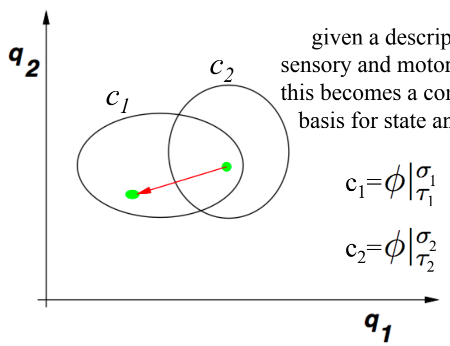
$$\boldsymbol{\kappa}_3 = \mathbf{J}_3^\# \Delta\phi_3$$

$$\boldsymbol{\kappa}_{3\triangleleft 2} = \mathbf{J}_2^\# \Delta\phi_2 + (\mathbf{I}_3 - \mathbf{J}_2^\# \mathbf{J}_2)\boldsymbol{\kappa}_3$$

$$\boldsymbol{\kappa}_{3\triangleleft 2\triangleleft 1} = \mathbf{J}_1^\# \Delta\phi_1 + (\mathbf{I}_3 - \mathbf{J}_1^\# \mathbf{J}_1)\boldsymbol{\kappa}_{3\triangleleft 2}$$

UMassAmherst

Tiling the State Space with Skills



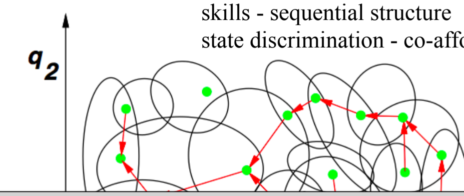
given a description of sensory and motor resources this becomes a combinatoric basis for state and action

$c_1 = \phi|^{\sigma_1}_{\tau_1}$

$c_2 = \phi|^{\sigma_2}_{\tau_2}$

UMassAmherst

Tiling the State Space with Skills



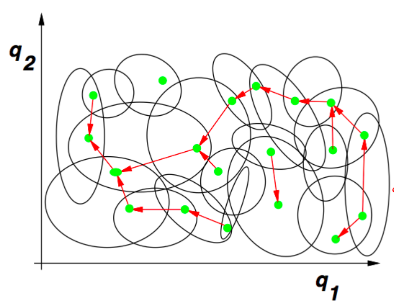
skills - sequential structure
state discrimination - co-affordances

some of these are referenced to stimuli in the environment that move in (semi)rigid groups … "objects," "rooms," etc

joint distributions/graph homomorphisms convey important context information

UMassAmherst

Tiling the State Space with Skills

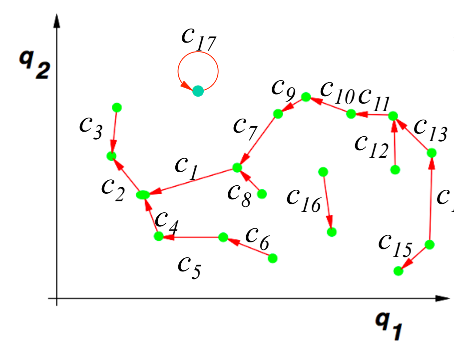


skills - sequential structure
state: [$\gamma_1 \ldots \gamma_n$]

"objects": transition dynamics

UMassAmherst

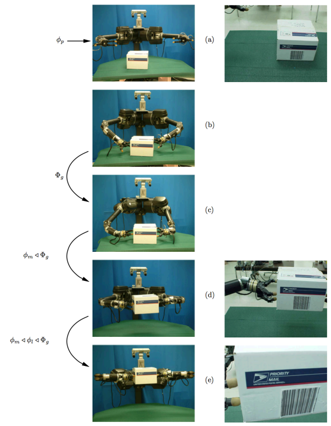
Tiling the State Space with Skills



LMT 1984
Burridge 1999
Tedrake 2009

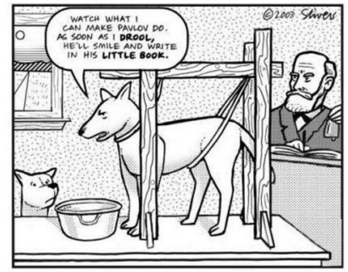
UMassAmherst

sequential, multi-objective control



UMassAmherst

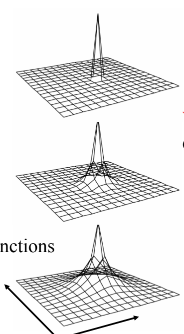
Conditioned Response



Pavlov, I. P. (1927), "Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex," Translated and Edited by G. V. Anrep. London: Oxford University Press.

UMassAmherst

A Computational Model for Conditioned Response



value functions

actions

states

value functions - an abstraction of the potential field

Reinforcement Learning - *value* iteration

- "diffusion" processes
- *curse of dimensionality* diminished by exploiting neurological structure

UMassAmherst

Markov Decision Processes

$M = < S, A, \Psi, P, R >$

S: set of system states

A: set of available actions

$\Psi \subseteq S \times A$ subset of actions allowed from each state

$P : S \times A \times S \rightarrow [0,1]$ probability that (s_k, a_k) transitions to state s_{k+1}

$R : S \times A \rightarrow \mathbb{R}$ real-valued reward for (state, action) pair

UMassAmherst

The Bellman Equation

Define a policy, $\pi(s, a)$, to be a function that returns the probability of selecting action $a \in A$ from state $s \in S$

the value of state s under policy π, denoted $V_\pi(s)$, is the expected sum of discounted future rewards when policy π is executed from state s,

$$V^\pi(s) = E_\pi \{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s \}$$

$0.0 < \gamma \leq 1.0$ represents a discounting factor per decision, and scalar r_t is the reward received at time t.

Laboratory for Perceptual Robotics – Department of Computer Science 17

UMassAmherst

The Bellman Equation

$$V^\pi(s) = E_\pi \{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s \}$$

$$= E_\pi \{ r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_t = s \}$$

$$= \sum_a \pi(s, a) \sum_{s'} P^a_{ss'} \ [R^a_{ss'} + \gamma E_\pi \{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_t = s' \}]$$

$$= \sum_a \pi(s, a) \sum_{s'} P^a_{ss'} \ [R^a_{ss'} + \gamma V^\pi(s')]$$

Laboratory for Perceptual Robotics – Department of Computer Science 18

UMassAmherst

The Bellman Equation

$$V^*(s) = \max_\pi V^\pi(s)$$

$$\boxed{V^*(s) = \max_a \sum_{s'} P^a_{ss'} \ [R^a_{ss'} + \gamma V^*(s')]}$$

Laboratory for Perceptual Robotics – Department of Computer Science 19

UMassAmherst

Reinforcement Learning: Q-Learning

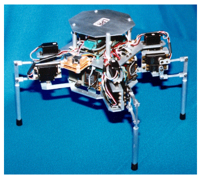
$$Q^\pi(s,a) = \sum_{s'} P^a_{ss'} [R^a_{ss'} + \gamma V_k(s')]$$

$$Q(s,a)_{k+1} \leftarrow Q(s,a)_k + \alpha \, [r(s') + \gamma \max_a Q(s',a) - Q(s,a)]$$

Laboratory for Perceptual Robotics – Department of Computer Science 20

UMassAmherst

Example: Learning to Walk (ca. 1996)



THING Quadruped
- four coordinated robots
- 2^{13} states × 1885 actions

Resource Model

- sensor resources -
  - configuration of legs {0123}
  - configuration of body (x,y,θ)
- effector resources -
  - configuration of legs {0123}
  - configuration of body (x,y,θ)
- control types -
  - moment control
  - kinematic conditioning

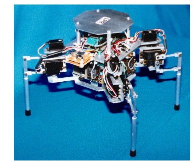
UMassAmherst

Example: Walking Gaits

13 controllers

moment control $\Phi_{1a}^{abc} \; abc \in \{0123\}$

kinematic conditioning $\Phi_{2\varphi}^{0123}$



total of 1885 concurrent control options

discrete events:

$p_0 \leftarrow \Phi_*^{012}$ $\quad p_3 \leftarrow \Phi_*^{013}$

$p_1 \leftarrow \Phi_*^{023}$ $\quad p_4 \leftarrow \Phi_\varphi^{0123}$

$p_2 \leftarrow \Phi_*^{123}$

UMassAmherst

Example: Behavioral Logic for Development

propositions that constrain patterns of discrete events in the dynamical system

Platform stability constraints
- at least 1 of 4 stable tripod stances to be true at all times

$$p_0 \vee p_1 \vee p_2 \vee p_3$$

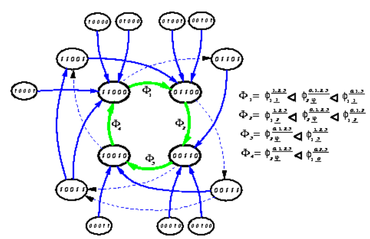
- kinematic constraints

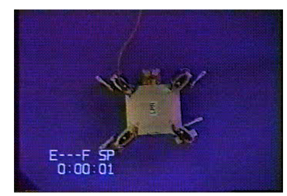
$$\neg(p_0 \wedge p_1) \wedge \neg(p_2 \wedge p_3)$$

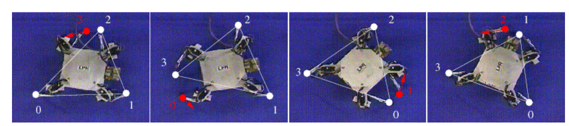
reduced model:
- 32 states × 157 actions
- reduced by 99.94 %

UMassAmherst

Example: ROTATE schema



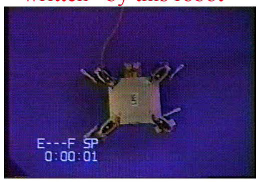




UMassAmherst

Transfer

"written" by this robot



ported to this robot

UMassAmherst

Implications of Developmental Hierarchy

native control
basis primitives

$\mathcal{A}_0 = \{\phi_1, \phi_2, \phi_3\}$

stage 1 $\downarrow$ $\mathcal{C}_1 = \{\phi_1, \phi_2\}$

$\mathcal{A}_1 = \{\phi_1, \phi_2, \phi_3, \underline{\text{ROTATE}}\}$

stage 2 $\downarrow$ $\mathcal{C}_2 = \{\phi_1, \phi_2, \text{ROTATE}\}$

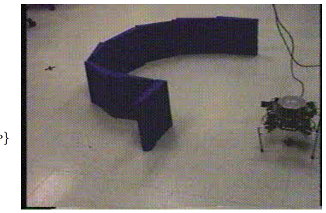
$\mathcal{A}_2 = \{\phi_1, \phi_2, \phi_3, \text{ROTATE}, \underline{\text{STEP}}\}$

stage 3 $\downarrow$ $\mathcal{C}_3 = \{\phi_1, \phi_2, \text{ROTATE}, \text{STEP}\}$

$\mathcal{A}_3 = \{\phi_1, \phi_2, \phi_3, \text{ROTATE}, \text{STEP}, \underline{\text{WALK}}\}$

stage 4 $\downarrow$ $\mathcal{C}_4 = \{\phi_3, \text{WALK}\}$

$\mathcal{A}_4 = \{\phi_1, \phi_2, \phi_3, \text{ROTATE}, \text{STEP}, \text{WALK}, \underline{\text{NAVIGATE}}\}$



fini