Some comments on control of robots (and people)

Various issues

Everything is optimization

The general policy is large

Model reductions

Prescriptive vs descriptive models

Andy Ruina, Mechanics, Cornell U.

N$F

Many of us seek to

make or **understand**

a

controller or **control system**

of a

robot or **animal/person**

to

not fall down, **move desired ways or places, or use little energy**.

A controller/policy is a map from sensors to actuators.

$$\mathbf{u}(t) = F\left\{\mathbf{s}(t'; -\infty < t' \leq t)\right\}$$

$\mathbf{u}(t)$ = the $n_{\text{actuators}}$ actuations at time t;

$\mathbf{s}$ = the m_{sensors} sensor readings as functions of time, up to t;

F = the controller.

How big/complex is it, generally?

$$n_{\text{table}} = n_{\text{actuators}} * n_d^{m_{\text{sensors}} * n_{\text{numb}}}$$

n_{numb} = the number of numbers for one scalar function
(*e.g.*, number of coefficients in Taylor series)

n_d = discretization of real numbers for a look up table (# bins).

Like, say, 10 google numbers in a look up table.

Finding the best controller (the best policy)
is optimizing, say, 10 google numbers.

$$n_{\text{table}} = n_a * n_d^{m_s * n_f}$$

Some problems:

1) Google is big.

   "*The curse of dimensionality*"

2) You have to decide what best/good is.

Repeating:

1) Finding a good approximation of a best policy means finding lots of numbers.

2) Deciding what is good/best is a challenge.

Most of us know these things, but don't (are afraid to) think about them. So

We

make up simplified control rules and, maybe,

try to make them good by this or that measure.

Simplifications

1) Forget old sensor readings, instead base control on present estimate of state.

$$\mathbf{u}(t) = F\{\hat{\mathbf{x}}\}$$

$\mathbf{u}(t)$ = the $n_{\text{actuators}}$ actuations at time t;

$\hat{\mathbf{x}}$ = estimate of the n_{state} state variables.

$$\Rightarrow n_{\text{table}} = n_{\text{actuators}} * n_d^{n_{\text{state}}}$$

Now it's only 10^11 numbers or so.

(Separation principle)

Simplifications (cont'd)

2) Model reduction/simplification.

Reduce number of state variables.

Assume hierarchical control.

Reduce number of basic actuations.

Make model or control "sparse"
(neglect lots of couplings.)

3) Guess a simple form for the control (only a few parameters, not billions).

4) Pick a simple-to-evaluate proxy for what you want as an "objective" function (something that takes little simulation, not millions of steps, to evaluate).

“Descriptive” vs “Prescriptive” Model Reduction

Descriptive: Let the system move any which way with any controller, but only pay attention to a reduced set of states and a reduced set of control actions (example on next page).

Prescriptive: Use controls to make the system move particular ways that increase the accuracy of a reduced model.

What's wrong with the prescriptive approach?

Overly restrictive.

Makes robot stiff.
Prevents dual tasking.
Limits range of strategies.

What's wrong with the descriptive approach?

Might not be accurate
Intrinsically adds noise.
One has to decide what to do with those internal DoFs.

Commuting Diagram

Broad point:

Think:

What would I optimize if I had infinite computing power available?

Given my skills and available computation and time, does my approach make the best approximation of that?