Optimization-Inspired Control Policies for Analytically and Computationally Intractable Systems

Christian Hubicki and Jonathan Hurst

Dynamic Robotics Laboratory
Oregon State University
Corvallis, Oregon

I. Motivation

We propose a method for synthesizing optimal controllers for analytically and computationally intractable dynamic systems. It is difficult to create controllers for legged systems for two primary reasons: a lack of closed-form solution (analytically intractable) and high dimensionality (computationally intractable). For example, there is no closed-form solution to the linearly elastic SLIP model so it is analytically intractable. Optimization inspiration, a counterpart to bio-inspiration, synthesizes controllers by observing patterns in a sampling of optimized controllers. The goal of optimization inspiration is to test and inspect these discovered patterns in order to uncover analytical insights.

II. State of the Art

There are a few approaches that are related to optimization inspiration of control. Dynamic Programming has been used to numerically compute globally optimal solutions [2]. Significant work has also been published specifically observing notable features of optimized gaits [6]. Optimization inspiration loosely follows the model of bio-inspiration, where observations of biological systems are used to find candidate strategies for interacting with the environment [3].

III. Discussion of Optimization-Inspired Control Policies

What does "Optimization-Inspired" mean?
Optimization-inspiration is a method of inspecting a small number of optimized policies to synthesize a generalized optimal control policy. Finding patterns in the optimal controllers can yield analytical insights into controlling the general system. Put more simply, it is the synthetic counterpart to bio-inspiration.

What challenges does the Optimization-Inspired approach overcome?
Optimization-inspired control policies, like more brute-force methods, require little-to-no analytical insight into the system to work. In fact, analytical insight is the goal of the process. Unlike brute-force methods, this approach randomly samples the state and parameter space so the burden of dimensionality is significantly alleviated [5].

What is the process of Optimization Inspiration?
The process is illustrated in Figure 1 and described below.

1) Exploration Set: Select a small sample set of points in the system's state space ($\underline{x}$) or parameter space (single parameter r in the visualized case).

This work is supported by the school of Mechanical, Industrial & Manufacturing Engineering at Oregon State University.

The authors can be contacted by email at:
hubickic@engr.oregonstate.edu

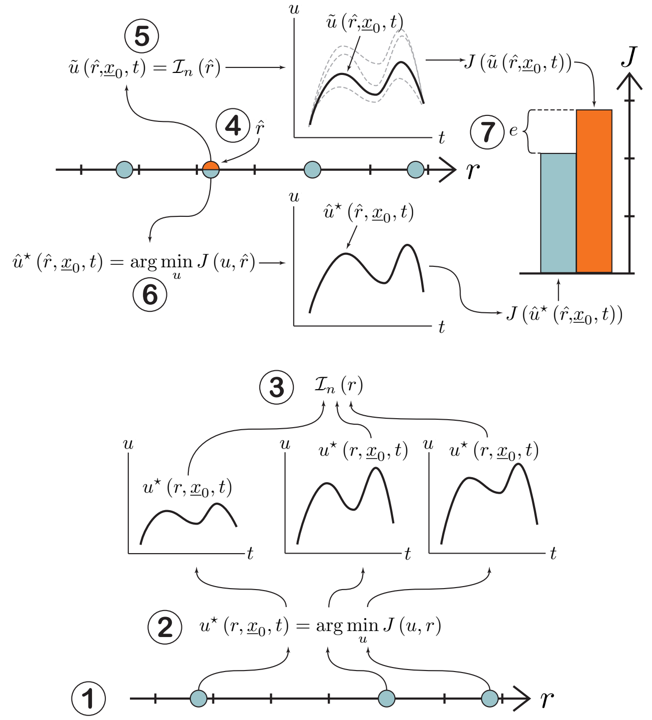

Fig. 1. The process of optimization inspiration, broken into seven steps, illustrated for a simple case: a system with a single time-varying input, $u(t)$, and a one-dimensional parameter space representing parameter, r.

2) Optimize: Using an optimization method of choice, minimize the cost function (J) to generate optimal controllers for each of these points.
3) Inspiration: Inspect these resulting controllers ($u^\star$) for correlations in relation to their respective locations in state/parameter space and generate an inspired pattern/heuristic ($\mathcal{I}_n$).
4) Test Set: Select another small sample set of points in the system's state space or parameter space ($\hat{r}$) for hypothesis testing.
5) Synthesize Controllers: Synthesize candidate controllers for each point in the hypothesis testing set by interpolating/extrapolating the candidate pattern ($\tilde{u}(\hat{r}, \underline{x}_0, t) = \mathcal{I}_n(\hat{r})$). These controllers are considered "optimization-inspired". Evaluate the cost of these optimization-inspired controllers ($J(\tilde{u}(\hat{r}, \underline{x}_0, t))$).
6) Optimize: Using the same optimization method as (2), generate optimal controllers ($\hat{u}^\star$) for each point in the hypothesis testing set.
7) Compare and Evaluate: If the error between the performance

of the candidate controller and the optimized controller ($e = J(\tilde{u}(\hat{r}, \underline{x}_0, t)) - J(\hat{u}^\star(\hat{r}, \underline{x}_0, t)))$ is small, it supports the hypothesis that the pattern is generalizable. Failure to match performance refutes the hypothesis and it needs to be revised or rejected. Consistent and close approximation of hypothesis-test-set performance by the optimization-inspired controllers is a strong indication of a globally applicable control policy.

What happens if no pattern is successfully tested?
If no successfully-tested pattern is found, it is possible that there is no general optimal policy, or that our approach simply did not find any that do exist. In this case, we are left only with the sparse sampling of optimized policies. A different control approach which populates the parameter space with greater density of optimized policies, such as dynamic programming, may be necessary.

How is optimization-inspired control different from using trajectory libraries [1] or memory-based dynamic programming [2]?
The optimized trajectories are not explicitly stored in the controller. The deliverable for the optimization-inspired approach is a description of the successfully-tested pattern or correlation.

Why use a pattern/correlation over a trajectory library?
A minor point is that interpolating using a pattern/correlation is more likely to yield optimal performance than a nearest-neighbor approximation using a trajectory library. Further, it is likely that fewer samples are required to identify and test a pattern than to provide sufficient optimal trajectories for a library yielding generally optimal performance over the entire state/parameter space. Also, with a parameter space properly represented, the optimization-inspired control policy can be directly implemented on similar robots of differing parameters. More significantly, a pattern or correlation is more instructive than a table. Important quantities or functions used in approximating the pattern may be meaningful in terms of the application. It is in this form where optimization-inspired control policies can lead to analytical insights.

What is meant by "parameter space"?
In this context, a parameter is defined as any quantity in the system dynamics or cost function that does not vary with time or state. The parameter space represents the scope of variation for any parameter which is meaningful to vary. The spring constant for an elastic system would be a fitting example. A generalized control policy for this one-dimensional parameter space yields an optimal control policy for any spring constant within specified bounds.

What is the point of having a varied parameter in the cost function?
Cost function parameters can produce performance tradeoffs. Defining the parameter space as a range of weighting factors for multiple performance metrics would allow an optimization-inspired control policy to generate a controller for any performance demand within this "tradeoff space". This is also a great method for generating a Pareto frontier for the system and its performance metrics. Conveniently, the process provides the control policy for any desired operating point along the Pareto frontier.

If an optimization-inspired control policy is successfully devised, do I have guarantees of optimality for untested points?
There is no analytical guarantee of optimality. Confidence in the policy is built on a statistical case using numerical experiments. These experiments prospectively test the hypothesis that the policy is globally optimal.

Why not use Lyapunov funneling or LQR trees [7] to cover the state space?
These are interesting techniques which may play a pivotal role in guaranteeing stability for high-dimensional nonlinear systems. That said, they are not designed to be optimal controllers outside of stability and robustness. Also, Lyapunov theory does not extend into parameter space as parameters are generally time-invariant.

How is this different than other work reporting observations of optimal gaits [6]?
Much of the work done highlighting notable features of optimal gaits is important. In fact, sheer observation is the first step in the process of creating optimization-inspired control policies. Optimization inspiration takes what is learned from these observations to directly synthesize controllers. Further, these synthesized controllers are prospectively tested against more optimizations to determine if the observation is meaningful for controlling the system.

Isn't this similar to how many biomechanists test hypotheses based on biological observations?
Yes. As the name implies, optimization inspiration has many similarities to bio-inspiration. It can be seen as an advantage of optimization-inspired control policies that they are in no way dependent upon the availability of relevant biological examples. Nature has not discovered or adopted optimal solutions to all problems. Furthermore, determining what a biological system is optimizing is a discipline in its own right [4]. With optimization inspiration the cost functions are explicitly defined. Simulation data is also vastly cheaper to obtain than biological data.

IV. Format

The authors would prefer a poster presentation.

V. Keywords

Optimization, Control, Inspiration

References

[1] C.G. Atkeson and B. Stephens. Multiple balance strategies from one optimization criterion. In *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pages 57 –64, 292007-dec.1 2007.
[2] C.G. Atkeson and B.J. Stephens. Random sampling of states in dynamic programming. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 38(4):924 –929, 2008.
[3] Monica A. Daley. Biomechanics: Running over uneven terrain is a no-brainer. *Current Biology*, 18(22):R1064 – R1066, 2008.
[4] Katja Mombaur, Anh Truong, and Jean-Paul Laumond. From human to humanoid locomotion - an inverse optimal control approach. *Autonomous Robots*, 28:369–383, 2010. 10.1007/s10514-009-9170-7.
[5] John Rust. Using randomization to break the curse of dimensionality. *Econometrica*, 65(3):pp. 487–516, 1997.
[6] M. Srinivasan. Fifteen observations on the structure of energy-minimizing gaits in many simple biped models. *Journal of the Royal Society*, 54(8):74–98, June 2010.
[7] Russ Tedrake, Ian R. Manchester, Mark Tobenkin, and John W. Roberts. Lqr-trees: Feedback motion planning via sums-of-squares verification. *Int. J. Rob. Res.*, 29:1038–1052, July 2010.