Policy Decomposition

Ashwin Khadke

Hartmut Geyer

Optimal Control is a formulation for posing the problem of designing controllers for dynamical systems whereby the desired long-term behavior of the system is expressed using a cost function. The objective is to compute a policy, i.e. a mapping from state of the system to control inputs, that minimizes the cost function. The complexity of computing optimal policies over the full state-space grows exponentially with the dimensionality of the state. Several approximation methods relying on either local search or state-space reduction have been proposed to alleviate this 'curse of dimensionality' and make synthesis of optimal controllers tractable for complex systems.

Local search methods such as iLQG [1], DDP [2] focus on the system behavior close to a reference motion and iteratively update both, the control as well as the reference motion, resulting in locally optimal policies. A global controller can be obtained by combining many such local policies but it is unclear how well the global controller approximates the true optimal policy. State-space reduction methods design control policies as a function of some lower-dimensional features of the state. These features are either hand-designed , or obtained by minimizing some projection error. However, these methods neither reason about nor predict the effect of these simplifications on the closed-loop behavior of the system. For linear systems, [3] reason about the closed-loop behavior in finding lower-dimensional representative systems. Methods that reason about the suboptimality of the resulting policies in finding suitable reductions for nonlinear systems are missing.

We introduce Policy Decomposition, a reduction method with an explicit suboptimality measure. Policy Decomposition proposes strategies to decouple and cascade the process of computing control policies for different inputs to a system (Fig. 1(a)). An intuitive abstraction to represent a strategy is using an input-tree (Fig. 1(b)). Each node in the input-tree (except the root) represents a subsystem. Subsystems that lie on the same branch are in a cascade and policies for inputs lower in the cascade influence the policies for inputs higher-up. Subsystems that lie on different branches are decoupled for the sake of policy computation. Based on the strategy/input-tree the control policies are a function of only a subset of the entire state of the system, and can be obtained by solving lower-dimensional optimal control problems leading to reduction in policy compute times.

We introduce the error between value functions of control policies obtained with and without decomposing to assess the suboptimality of a decomposition strategy. This error cannot be obtained without knowing the true optimal control, and we estimate it based on LQR and DDP approximations. The LQR based estimate can be computed in minimal time whereas the DDP based estimate is more accurate. Using a cart pole, a 3-link balancing biped, and N-link planar manipulators, we show the proposed method can identify decomposition strategies that result in substantial reduction in policy computation times while sacrificing little in closed-loop performance. To systematically search through the possible policy decompositions for a system, we explore the use of search methods such as Genetic Algorithm (GA) and Monte-Carlo Tree Search (MCTS) to quickly find promising decompositions. Our experiments suggest that Policy Decomposition is viable alternative for simplifying optimal control synthesis for complex systems.

References

[1] E. Todorov and W. Li, "A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *American Control Conference*. IEEE, 2005, pp. 300–306.

[2] D. Q. Mayne, "Differential dynamic programming–a unified approach to the optimization of dynamic systems," in *Control and Dynamic Systems*. Elsevier, 1973, vol. 10, pp. 179–254.

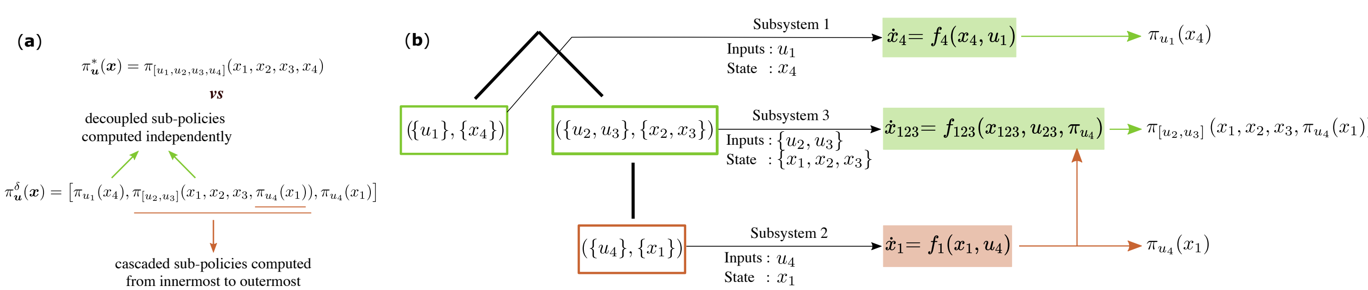

Fig. 1. (**a**) Concept of policy decomposition shown for a fictive system with four states and four inputs. (**b**) An input-tree that describes the policy decomposition described in (**a**) is depicted (**left**). The three resulting subsystems are shown (**mid**). Policies for u_1 and u_4, i.e. the inputs at the leaf nodes, are obtained first by independently solving the optimal control problems for subsystems 1 and 2 respectively. Policies for inputs u_2 and u_3 are computed jointly by solving the optimal control problem for subsystem 3. The resulting policies for different inputs are a function of different state variables (**right**).

[3] N. Xue and A. Chakrabortty, “Optimal control of large-scale networks using clustering based projections,” *arXiv preprint arXiv:1609.05265*, 2016.