ENSTA IP PARIS

CSC_5RO01_TA

Hybrid Optimal Control

Lecture 5: Algorithms & Implementation

Mario Gleirscher (v0.1, 19/12/25)

3A/Master Course, 2 ETCS, 2025/26

Learning Goals

Questions:

1. How can we implement dynamic programming (DP) algorithms for computing tabular controls?

At the end of this lecture, you will ...

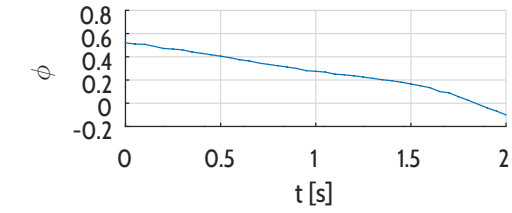
1. have learned how to implement a rudimentary variant of value iteration for discrete dynamic programming (DDP).

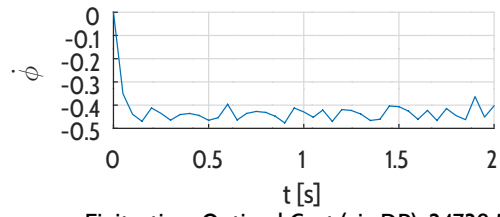
Important Aspects when Implementing Value Iteration

1. Discretise the dynamics.
2. Identify the scope of the discrete state and input spaces.
3. Identify parameter resolutions for state and input spaces and the time step.
4. Identify the data structures for the state space, value function, and tabular control law.
5. Choose an lookup and interpolation scheme (e.g., nearest neighbour).
6. Choose a numerical integration scheme (e.g., Euler, Runge-Kutta).
7. Implement a simulator monitoring relevant events (e.g., jump guards of a hybrid automaton, scope violations).
8. Add an appropriate visualisation.

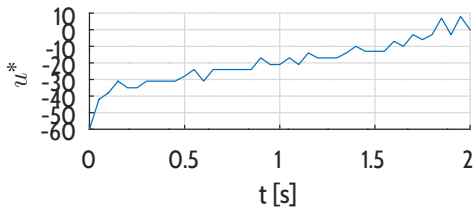
These steps will heavily depend on the chosen platform (e.g., C, C++, Python, Matlab).

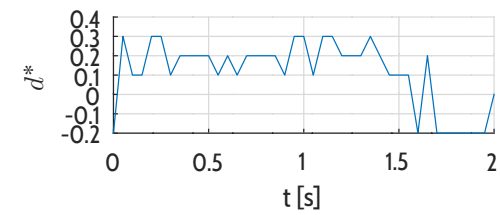
Differential Game for an Inverted Pendulum

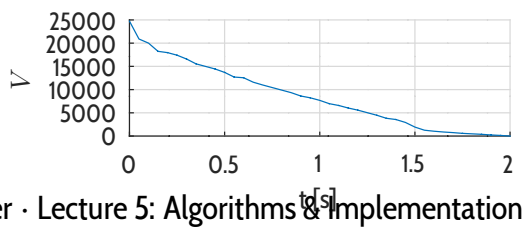


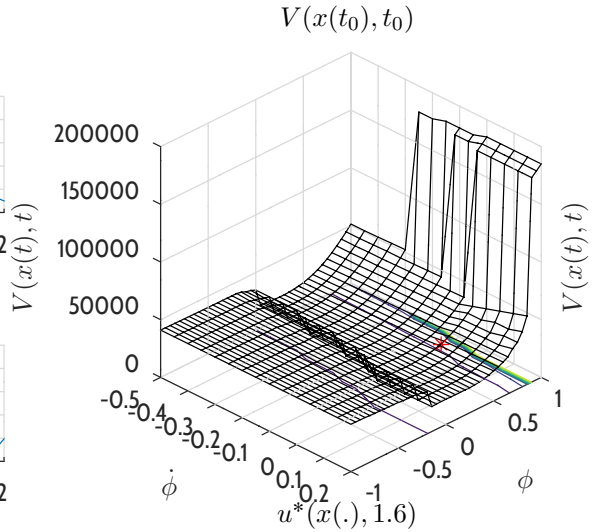


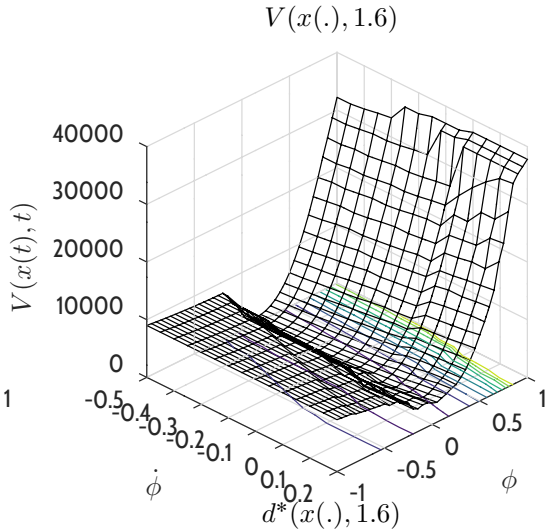
Finite-time Optimal Cost (via DP): 24739.52

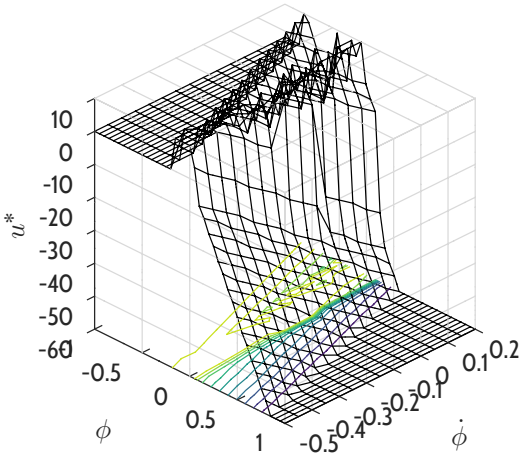


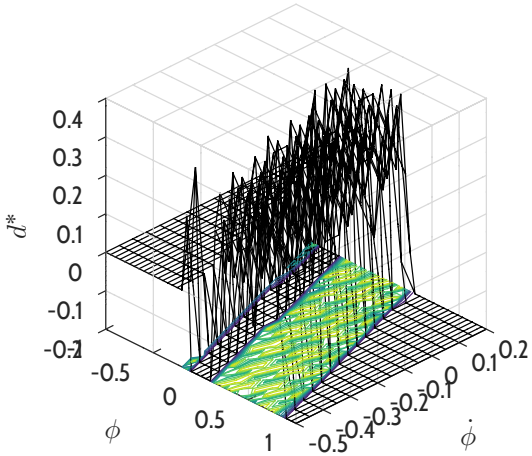












- DDP algorithms, such as value iteration, can generate tabular control laws.
- Exhaustive variants of such algorithms require computation time and storage space exponential in the number of state variables.
- In digital control, such laws can also be employed in a piece-wise constant control regime, where in state $\mathbf{x}$, each input $\mathbf{u}(\mathbf{x})$ is applied continuously for the length of a sampling period δt.