Trajectory Following Optimization Methods

- Player strategies are generated through solutions to ordinary differential equations.
- These differential equations balance the rate of cost accumulation with penetration rate for an appropriate descent function.
- We derive these strategies, \( \hat{f} \), from a function \( \hat{W}_0 \).

\[
\dot{x} = f(x, u, v) \\
\dot{u} = \hat{f}(x, u, v)
\]

Min-Max Differential Games

- The solution requires that we form the differential game \( H \) function.

\[
H = \lambda_0 f_0(x, u, v) + \lambda^T f(x, u, v)
\]

- Perform the following possibly nonlinear min-max optimization at each point along the state trajectory:

\[
\min_{u \in U} \max_{v \in V} H(x, u, v, \lambda)
\]

We will derive the Efficient Cost Descent Algorithm

- We may avoid solving the above min-max optimization problem at each point of the state trajectory.
- This algorithm is not stiff, therefore easy to implement numerically.
- Use of the trajectory following methods provides for “on-line” implementation of player strategies.

Linearized Inverted Pendulum

State trajectories and \( v \) – reachable sets

How do we design a robust algorithm that drives the state to the target?

- Merge the trajectory following method with differential game theory.
- This results in the Efficient Cost Descent Algorithm

\[
\dot{u} = - \left[ \frac{\exp(-\alpha t)}{\epsilon} + \beta (||\dot{g}||) \right] \dot{g} - \left( \frac{\partial g}{\partial u} \right)^{-1} \frac{\partial g}{\partial x} \dot{x} + \dot{u}_v
\]

- The algorithm is efficient (fast)
- Numerically tractable (ode integrator)
- Second time scale quickly decays with the exponential term
- Behaves like Newton’s method.
- Roughly 6700 times faster than SPMCD

Trajectory Following Algorithms for Differential Games

Dale B. McDonald and Walter J. Grantham, School of Mechanical and Materials Engineering, Washington State University