Verification of Stochastic Systems

Motivation

- As robots move from factory floors to more demanding environments, they will have to cope with increasingly complex uncertainty.
  - Perceptual uncertainty from stereo vision or cluttered environments.
  - Dynamical uncertainty from rough terrain, wind gusts, or grasping soft fabrics.
- Classical approach: robust control.
  - If my uncertainty stays bounded in a certain region, then I am guaranteed to reach my goal.
- Problems: high-tailed noise, conservative due to worst-case planning.
- Goal: develop algorithms to deal with explicitly-modeled uncertainty.

Background

- 1965: Kushner provides Lyapunov-like techniques for obtaining probabilistic guarantees about trajectories of Markov chains; paper includes several handworked examples, but he doesn’t have the computational machinery to develop general algorithms.
- 2001: Prajna et al. provide an algorithm for bounding trajectories of switching systems with Gaussian noise. They use sum-of-squares programming on Martingales, but cannot handle noise at the origin and use a basis that leads to conservative results.
- Our contribution: we combine Prajna’s algorithm with Kushner’s theory to handle noise at the origin. We also work in a basis that provides much tighter bounds at the expense of more difficult computations.

Martingales

- A non-negative function V of the state x is a c-martingale if E[V(x)] ≤ c.
- Theorem (Kushner 1965): Suppose that V is a martingale in the region where \( V(x) < \rho \). Then the probability that x leaves the region \( \{ V(x) \leq \rho \} \) before time \( T \) is at most \( \exp(-2cT) \). Time-varying version also holds as long as \( V \) is a continuous function of time.

Use in controller synthesis

- A single martingale \( V \) will yield bounds for an entire family of controllers (see figure to right).
- We can use this bound as a proxy for controller quality and optimize our choice of controller against the provided bound.
- Repeating this process is called DK iteration.

Martingales for Gaussian Systems

Calculating the Expected Derivative

Consider a system with Gaussian noise: \( dx(t) = f(x(t))dt + g(x(t))dw(t) \), where \( dw(t) \) is a vector of i.i.d. Wiener processes.

\[ E[V(x)] = \frac{1}{T} \int_0^T g(x(t))^\top S(x(t)) g(x(t)) dt \]

Why \( E[V(x)] \leq c \) instead of \( E[V(x)] \leq 0 \)?

- Consider the equation \( dV(x) = -cV(x)dt + dV(x) \) (above)
- Trajectory decays towards origin, then bounces around
- \( E[V(x)] \leq 0 \) will be too be too strict in this case
- Relaxing to \( E[V(x)] \leq c \) allows us to handle noise at the origin (improvement over previous work)

Relaxing to a Polynomial Condition

- We want to check if \( \rho e^{-ct} \) is bounded for polynomial \( \rho \) and \( c \).
- Reasons: \( \rho(x) \leq 2^{\kappa+2} \rho(0) \).
- Note that \( \rho(x) \leq e^{\kappa x} \) by convexity.
- Sufficient condition: \( \rho(x) \leq 2^{\kappa+2} \rho(0) \).
- This is a polynomial condition, and Schur complements can be used to make it linear in the decision variable. We then obtain the real result:

Solutions

- We need to find an initial feasible point.
- In the step where we fix \( \rho \), we used a different objective function (maximizing \( \rho \) does not make sense as \( \rho \) is a decision variable).
- Modern numerical optimizers typically give solutions slightly outside the feasible region. Boxer-Neumaier can cause these errors to accumulate and lead to numerical instability.

- Initialize with an S matrix for the linearized system and a small value of \( \rho \).
- First minimize \( \rho \), then maximize \( c \), and take the average of the two solutions (this approach to find a point close to the middle of the feasible region).
- After each maximization step, find a feasible point whose objective value is only slightly less than the value obtained from the maximization. For instance, if the maximization returns \( \rho = 2.2 \), then find a feasible point with the added constraint \( \rho < 2.28 \).

Optimization Techniques for Choosing a Martingale

- Since our bound on the failure probability has an \( e^c \) term in the denominator, we will try to make \( \rho \) as large as possible.
- Note that if we fix \( \epsilon \) and \( \lambda \), the constraint is linear in \( \rho \) and \( \lambda \). Likewise, if we fix \( \rho \) and \( \lambda \), the constraint is linear in \( \epsilon \) and \( \lambda \).
- Optimization strategy: First fix \( \epsilon \) and \( \lambda \) and optimize \( \rho \) and \( \lambda \), then fix \( \rho \) and \( \lambda \) and optimize \( \epsilon \) and \( \lambda \). But, there are a few issues to resolve.

Issues

- Our approach to find the 7-room case in under 15 minutes.
- Our approach solves the 7-room case in under 6 hours.
- Our approach can handle at least the 10-room case, and scales polynomially with dimension.