Measures and LMI for space launcher robust control validation

Didier HENRION
LAAS-CNRS & FEL-ČVUT

Martine GANET-SCHOELLER
EADS ASTRIUM Space Transportation

Samir BENNANI
ESTEC European Space Agency

Spring 2012

Outline

1. Occupation measures for validation
2. Generalized moment problem and LMI relaxations
3. Application on ACS1 benchmark
4. Discussion

From Cauchy to Liouville

Dynamical system described by nonlinear ODE

$$\dot{x} = f(x)$$

with Lipschitz vector field f has a unique solution (trajectory) x_t starting from a given initial condition $x_0 \in \mathbb{R}^n$

Suppose x_0 is not known exactly: think of it as a random variable modeled by a probability measure μ_0

At time t, state x_t is also modeled by a probability measure μ_t whose time evolution is captured by a linear PDE

$$\frac{\partial \mu_t}{\partial t} + \sum_{i=1}^{n} \frac{\partial (f \mu_t)_i}{\partial x_i} = 0$$

called Liouville's transport or advection equation

Occupation measure

Given a compact set $X \subset \mathbb{R}^n$, define the occupation measure

$$\mu(X) = \int_0^T \mu_t(X)dt$$

which measures the time spent by the trajectory in X

Measure μ encodes the whole trajectory and satisfies the integral Liouville PDE

$$\text{div}\,(f\mu) = \mu_0 - \mu_T$$

and its variational formulation

$$\int_X Dv \cdot f d\mu = \int_X v d\mu_T - \int_X v d\mu_0$$

for all smooth test functions v supported on X

Dynamic optimization

We consider that our validation problem can be formulated as a polynomial dynamic optimization problem over trajectories

$$\begin{array}{rll} J = & \inf_{x(t)} & h_T(x(T)) + \int_0^T h(x(t))dt \\ & \text{s.t.} & \dot{x}(t) = f_k(x(t)), \quad x(t) \in X_k, \quad k = 1, 2, \ldots, N \\ & & x(0) \in X_0, \quad x(T) \in X_T, \quad t \in [0, T] \end{array}$$

with given polynomial dynamics $f_k \in \mathbb{R}[x]$ and costs $h, hT \in \mathbb{R}[x]$ defined on basic semialgebraic sets

$$X_k = \{x \in \mathbb{R}^n \,:\, g_{kj}(x) \geq 0, \quad j = 1, 2, \ldots, N_k\}$$

for given polynomials $g_{kj} \in \mathbb{R}[x]$

The infimum is sought over absolutely continuous trajectories $x(t)$: this problem is infinite-dimensional, nonlinear, nonconvex !

Linear program on measures

..but it can be reformulated as a linear hence convex problem

$$J_\infty = \ \text{inf}_\mu \ \int h_T \, d\mu_T + \sum_k \int h \, d\mu_k$$
$$\text{s.t.} \quad \sum_k \int Dv \cdot f_k d\mu_k = \int v \, d\mu_T - \int v \, d\mu_0, \quad \forall v \in C^1(X)$$

where the unknowns are local occupation measures μ_k and the global occupation measure is given by

$$\mu = \sum_k \mu_k, \quad \mu(X) = T$$

Final time T, initial measure μ_0 and terminal measure μ_T may be given, or unknown (depending on the problem)

The problem is linear but still infinite-dimensional..

Outline

1. Occupation measures for validation
2. **Generalized moment problem and LMI relaxations**
3. Application on ACS1 benchmark
4. Discussion

Measures and their moments

So far we have formulated our validation problem as a dynamic polynomial optimization problem, and then we have linearized this problem to obtain an LP measure problem:

$$J_\infty = \inf_\mu \sum_k \int c_k d\mu_k$$
$$\text{s.t.} \quad \sum_k \int a_{ki} d\mu_k = b_i, \quad \forall i$$

where our unknowns are a finite set of measures μ_k

The next step consists of manipulating each measure μ_k via its moments

$$y_{k\alpha} = \int x^\alpha d\mu_k, \quad \forall \alpha \in \mathbb{N}^n$$

Moments and LMI relaxations

Our LP measure problem becomes an LP moment problem:

$$
\begin{array}{rll}
J_\infty = & \inf_y & \sum_k \sum_\alpha c_{k\alpha} y_{k\alpha} \\
& \text{s.t.} & \sum_k \sum_\alpha a_{ki\alpha} y_{k\alpha} = b_i, \quad \forall i
\end{array}
$$

where our unknowns are a finite set of infinite-dimensional sequences y_k representing our measures

The constraint that y_k contains the moments of measure μ_k can be formulated as an infinite-dimensional LMI problem, and we can build a hierarchy of finite-dimensional LMI relaxations:

$$
\begin{array}{rll}
J_d = & \inf_y & c^T y \\
& \text{s.t.} & Ay = b \\
& & F(y) = \sum_k \sum_\alpha F_{k\alpha} y_{k\alpha} \succeq 0
\end{array}
$$

such that $J_d \leq J_{d+1}$ and $\lim_{d\to\infty} J_d = J_\infty = J$

Outline

1. Occupation measures for validation
2. Generalized moment problem and LMI relaxations
3. **Application on ACS1 benchmark**
4. Discussion

Attitude Control System (ACS)

Orbital phase: after main propulsion of upper stage

Composed of manoeuvres performed by ACS for e.g. payload separation, distancing, non-pollution, passivation, de-orbiting, stage re-ignition

SAFE-V ACS benchmark studies effects of longitudinal spin and axis coupling

ACS1 1DOF simplified model

No time-delay, no pulsation width modulator

Double integrator

$$\begin{bmatrix} \dot{x}_1 \\ I\dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ u \end{bmatrix}$$

with torque control

$$u(x(t)) = \text{sat}(K^T \text{dz}(x_r(t) - x(t)))$$

with state-feedback K, saturation (sat) and deadzone (dz) designed to follow reference signal $x_r(t)$

Verification problem: after a fixed time T, does the system state $x(T)$ reach a given subset X_T of the deadzone region, for all possible initial conditions $x(0)$ chosen in a given subset X_0, for reference signal $x_r(t) = 0$?

Piecewise affine model

Three cells with affine dynamics: linear regime

$$X_1 = \{x \,:\, |K^T x| \leq L\}, \quad f_1(x) = \begin{bmatrix} x_1 \\ -K^T x \end{bmatrix}$$

upper saturation regime

$$X_2 = \{x \,:\, K^T x \geq L\}, \quad f_2(x) = \begin{bmatrix} x_1 \\ -L \end{bmatrix}$$

lower saturation regime

$$X_3 = \{x \,:\, K^T x \leq -L\}, \quad f_3(x) = \begin{bmatrix} x_1 \\ L \end{bmatrix}$$

Objective function to be minimized
to find worst-case trajectory

$$h_T(x) = -x(T)^T x(T)$$

GloptiPoly code

```
% measures
mpol('x1',2); m1 = meas(x1); % linear regime
mpol('x2',2); m2 = meas(x2); % upper saturation
mpol('x3',2); m3 = meas(x3); % lower saturation
mpol('x0',2); m0 = meas(x0); % initial
mpol('xT',2); mT = meas(xT); % terminal

% dynamics on normalized time range [0,1]
% saturation input y normalized in [-1,1]
K = -[kp kd]/L;
y1 = K*x1; f1 = T*[x1(2); L*y1/I]; % linear regime
y2 = K*x2; f2 = T*[x2(2); L/I]; % upper sat
y3 = K*x3; f3 = T*[x3(2); -L/I]; % lower set
```

```
% test functions for each measure = monomials
% with d = given relaxation order
g1 = mmon(x1,d); g2 = mmon(x2,d); g3 = mmon(x3,d);
g0 = mmon(x0,d); gT = mmon(xT,d);

% unknown moments of initial and final measures
y0 = mom(g0); yT = mom(gT);

% input LMI moment problem
cost = mom(xT'*xT);
Ay = mom(diff(g1,x1)*f1)+...
     mom(diff(g2,x2)*f2)+...
     mom(diff(g3,x3)*f3); % dynamics

% trajectory constraints
X = [y1^2<=1; y2>=1; y3<=-1];
% initial constraints
X0 = [x0(1)^2<=thetamax^2, x0(2)^2<=omegamax^2];
% terminal constraints (in deadzone region)
XT = [xT'*xT<=epsilon^2];
% bounds on trajectory
B = [x1'*x1<=4; x2'*x2<=4; x3'*x3<=4];
```

```
% input LMI moment problem
P = msdp(max(cost), ...
         mass(m1)+mass(m2)+mass(m3)==1, ...
         mass(m0)==1, ...
         Ay==yT-y0, ...
         X, X0, XT, B);

% solve LMI moment problem
% using a general-purpose SDP solver (e.g. SeDuMi)
[status,obj] = msol(P)
```

Results

relaxation order d	1	2	3	4
upper bound J_d	$1.0 \cdot 10^{-5}$	$1.0 \cdot 10^{-5}$	$1.0 \cdot 10^{-5}$	$1.0 \cdot 10^{-5}$
CPU time (sec.)	0.2	0.5	0.7	0.9
number of moments	30	75	140	225

We see that the bound obtained at the first relaxation ($d = 1$) is not modified for higher relaxations

All initial conditions are captured in the deadzone region so the control law is validated

Outline

1. Occupation measures for validation
2. Generalized moment problem and LMI relaxations
3. Application on ACS1 benchmark
4. **Discussion**

Complexity

Computation burden does not depend too much on the number of cells modelling the nonlinearities (i.e. the number of measures) but it depends critically on the number of state variables

For a simple LMI

$$\begin{array}{ll} \inf_y & c^T y \\ \text{s.t.} & M_d(y_k) \succeq 0, \quad k = 1, 2, \ldots, K \end{array}$$

where $M_d(y_k)$ is the moment matrix of a measure μ_k of n variables at relaxation order d, the (worst-case) complexity for a primal-dual interior point algorithm grows in $O(Kd^{4n})$

Weak (linear) dependence on number of measures K
Strong (exponential) dependence on number of states n

Complexity

Real uncertain parameters can be handled as additional states (see technical report for ACS1 1DOF with uncertain inertia)

ACS1 3DOF model (quadratic nonlinearities, 10 states) can also be handled with these techniques:

relaxation order d	1	2	3	4
CPU time (sec.)	0.2	11.6	24.2	2640
number of moments	110	770	1430	5720

4th LMI relaxation is quite demanding, but in this case useful bounds for validation were already obtained at lower LMI relaxation orders

Extension to time-delay systems

Example of nonlinear ODE with one time-delay $\tau > 0$:

$$\dot{x}(t) = f(x(t)) + g(x(t-\tau)), \quad \forall t \in [0, T]$$

with boundary conditions

$$x(t) = \xi(t), \ \forall t \in [-\tau, 0]$$

where $\xi(t)$ is a given function recording the state history

Instead of transporting a probability measure $\mu_t(dx)$ supported on $X \subset \mathbb{R}^n$ from initial time $t = 0$ to terminal time $t = T$, we must transport the state history in an occupation measure $\mu_t(ds, dx)$ supported on $[-\tau, 0] \times X$ for $t \in [0, T]$

Extension to discrete-time systems

The probability measure $\mu_k(x)$ transported along nonlinear discrete-time system dynamics

$$x_{k+1} = f(x_k)$$

satisfies the discrete-time linear Liouville equation

$$\mu_{k+1}(X) = \int_{f^{-1}(X)} \mu_k(dx) = \int I_X(f(x))\mu_k(dx)$$

Moments of measure μ_{k+1} can be expressed linearly as functions of moments of measure μ_k as follows

$$\int v(x)\mu_{k+1}(dx) = \int v(f(x))\mu_k(dx)$$

for all test functions, e.g. $v(x) = x^\alpha$, $\alpha \in \mathbb{N}^n$

Conclusion

Main features of moment method:

- verification/validation formulated as nonlinear nonconvex infinite-dimensional functional optimization problem
- problem is linearized in measure space
- measures are handled via moments and hierarchy of LMIs
- SDP solver provides monotonically sequence of bounds
- models are piecewise polynomial, no use of LFT
- software readily available (GloptiPoly, SeDuMi)