Computational Issues in Nonlinear Dynamics and Control

Arthur J. Krener

ajkrener@ucdavis.edu

Supported by AFOSR and NSF

Typical Problems

- **Numerical Computation of Invariant Manifolds**

Typical Problems

- **Numerical Computation of Invariant Manifolds**
- **Numerical Solution of Optimal Control Problems**

Typical Problems

- **Numerical Computation of Invariant Manifolds**
- **Numerical Solution of Optimal Control Problems**
- **Numerical Solution of Differential Games**

Typical Problems

- **Numerical Computation of Invariant Manifolds**
- **Numerical Solution of Optimal Control Problems**
- **Numerical Solution of Differential Games**
- **Numerical Solution of Conservation Laws**

Typical Problems

- **Numerical Computation of Invariant Manifolds**
- **Numerical Solution of Optimal Control Problems**
- **Numerical Solution of Differential Games**
- **Numerical Solution of Conservation Laws**
- **Numerical Solution of Frances Byrnes Isidori PDEs**

Typical Problems

- **Numerical Computation of Invariant Manifolds**
- **Numerical Solution of Optimal Control Problems**
- **Numerical Solution of Differential Games**
- **Numerical Solution of Conservation Laws**
- **Numerical Solution of Frances Byrnes Isidori PDEs**
- **Numerical Solution of Kazantzis Kravaris PDEs**

Typical Problems

- **Numerical Computation of Invariant Manifolds**
- **Numerical Solution of Optimal Control Problems**
- **Numerical Solution of Differential Games**
- **Numerical Solution of Conservation Laws**
- **Numerical Solution of Frances Byrnes Isidori PDEs**
- **Numerical Solution of Kazantzis Kravaris PDEs**
- **Numerical Solution of Duncan-Mortenson-Zakai PDEs**

Typical Problems

- **Numerical Computation of Invariant Manifolds**
- **Numerical Solution of Optimal Control Problems**
- **Numerical Solution of Differential Games**
- **Numerical Solution of Conservation Laws**
- **Numerical Solution of Frances Byrnes Isidori PDEs**
- **Numerical Solution of Kazantzis Kravaris PDEs**
- **Numerical Solution of Duncan-Mortenson-Zakai PDEs**
- **Numerical Solution of H^∞ PDEs**

Typical Problems

- **Numerical Computation of Invariant Manifolds**
- **Numerical Solution of Optimal Control Problems**
- **Numerical Solution of Differential Games**
- **Numerical Solution of Conservation Laws**
- **Numerical Solution of Frances Byrnes Isidori PDEs**
- **Numerical Solution of Kazantzis Kravaris PDEs**
- **Numerical Solution of Duncan-Mortenson-Zakai PDEs**
- **Numerical Solution of H^∞ PDEs**

An Important Problem

Infinite Horizon Optimal Control

$$\min_{u(0:\infty)} \int_0^\infty l(x,u)\; dt$$

$$\dot{x} = f(x,u), \qquad x(0) = x^0$$
$$x \in \mathbb{R}^{n\times 1}, \qquad u \in \mathbb{R}^{m\times 1}$$

An Important Problem

Infinite Horizon Optimal Control

$$\min_{u(0:\infty)} \int_0^\infty l(x,u)\ dt$$

$$\dot{x} = f(x,u), \qquad x(0) = x^0$$
$$x \in \mathbb{R}^{n\times 1}, \qquad u \in \mathbb{R}^{m\times 1}$$

Optimal Cost and Optimal Feedback

$$\pi(x^0) = \min_{u(0:\infty)} \int_0^\infty l(x,u)\ dt, \qquad u^*(0) = \kappa(x^0)$$

An Important Problem

Infinite Horizon Optimal Control

$$\min_{u(0:\infty)} \int_0^\infty l(x,u)\ dt$$

$$\begin{aligned} \dot{x} = f(x,u), &\qquad x(0) = x^0 \\ x \in I\!R^{n\times 1}, &\qquad u \in I\!R^{m\times 1} \end{aligned}$$

Optimal Cost and Optimal Feedback

$$\pi(x^0) = \min_{u(0:\infty)} \int_0^\infty l(x,u)\ dt, \qquad u^*(0) = \kappa(x^0)$$

Hamiltonian, a function of x, u and a new variable $p \in I\!R^{1\times n}$

$$\mathcal{H}(p,x,u) \;=\; pf(x,u) + l(x,u)$$

An Important Problem

Infinite Horizon Optimal Control

$$\min_{u(0:\infty)} \int_0^\infty l(x,u)\ dt$$

$$\begin{aligned} \dot{x} = f(x,u), &\qquad x(0) = x^0 \\ x \in I\!R^{n\times 1}, &\qquad u \in I\!R^{m\times 1} \end{aligned}$$

Optimal Cost and Optimal Feedback

$$\pi(x^0) = \min_{u(0:\infty)} \int_0^\infty l(x,u)\ dt, \qquad u^*(0) = \kappa(x^0)$$

Hamiltonian, a function of x, u and a new variable $p \in I\!R^{1\times n}$

$$\mathcal{H}(p,x,u) \;=\; pf(x,u) + l(x,u)$$

Hamilton Jacobi Bellman Equations

$$\begin{aligned} 0 &= \min_u \mathcal{H}(\frac{\partial \pi}{\partial x}(x), x, u) \\ \kappa(x) &= \operatorname{argmin}_u \mathcal{H}(\frac{\partial \pi}{\partial x}(x), x, u) \end{aligned}$$

Stabilization by Optimization

Problem: Find a feedback $u = \kappa(x)$ so that the closed loop system is (locally) asymptotically stable around $x = 0$.

Solution: Choose a Lagrangian $l(x, u) \geq 0$ and solve the infinite horizon optimal control problem. Under suitable conditions the optimal feedback $u = \kappa(x)$ is stabilizing on some domain around $x = 0$ and this can be verified because the optimal cost $\pi(x) \geq 0$ is a Lyapunov function,

$$\frac{d}{dt}\pi(x(t)) = \frac{\partial \pi}{\partial x}(x(t))f(x(t), \kappa(x(t))) = -l(x(t), \kappa(x(t))) \leq 0$$

Classic Example: LQR

$$\boldsymbol{f(x,u) = Fx + Gu, \qquad l(x,u) = \frac{1}{2}\left(x'QX + u'Ru\right)}$$

Classic Example: LQR

$$f(x,u) = Fx + Gu, \qquad l(x,u) = \frac{1}{2}\left(x'QX + u'Ru\right)$$

Optimal Cost and Optimal Feedback

$$\pi(x) = \frac{1}{2}x'Px, \qquad \kappa(x) = Kx$$

The HJB equations reduce to a quadratic (algebraic Riccati) equation and a linear equation

$$0 = F'P + PF + Q - PGR^{-1}G'P, \qquad K = -R^{-1}G'P$$

Classic Example: LQR

$$f(x,u) = Fx + Gu, \qquad l(x,u) = \frac{1}{2}\left(x'QX + u'Ru\right)$$

Optimal Cost and Optimal Feedback

$$\pi(x) = \frac{1}{2}x'Px, \qquad \kappa(x) = Kx$$

The HJB equations reduce to a quadratic (algebraic Riccati) equation and a linear equation

$$0 = F'P + PF + Q - PGR^{-1}G'P, \qquad K = -R^{-1}G'P$$

Theorem: If $Q \geq 0,\ R > 0$, (F,G) stabilizable and $(Q^{1/2},F)$ detectable then there exist a unique nonnegative definite solution P to the Riccati equation and the feedback $u = Kx$ is asymtotically stable, i.e., all the poles of $F + GK$ are in the open left half plane.

Another Important Problem

Finite Horizon Optimal Control

$$\min_{u(0:T)} \int_0^T l(t,x,u)\ dt + \pi^T(x(T))$$

$$\begin{aligned} \dot{x} &= f(t,x,u) \\ 0 &= g(x(0), X(T)) \\ u(t) &\in \mathcal{U}(t,x) \end{aligned}$$

Another Important Problem

Finite Horizon Optimal Control

$$\min_{u(0:T)} \int_0^T l(t,x,u)\ dt + \pi^T(x(T))$$

$$\begin{aligned} \dot{x} &= f(t,x,u) \\ 0 &= g(x(0),X(T)) \\ u(t) &\in \mathcal{U}(t,x) \end{aligned}$$

Pontryagin Maximum Principle:

If $x^*(0:T),\ u^*(0:T)$ is optimal then there exists $p:[0,T]\to I\!R^{1\times n}$ such that

$$\begin{aligned} \dot{x}_i^* &= \frac{\partial \mathcal{H}}{\partial p_i}(t,p,x^*,u^*) \\ \dot{p}_i &= -\frac{\partial \mathcal{H}}{\partial x_i}(t,p,x^*,u^*) \\ u^* &= \mathrm{argmin}_{u\in\mathcal{U}(T,x^*)} \mathcal{H}(t,p,x^*,u^*) \\ \mathcal{H}(t,p,x,u) &= pf(t,x,u)+l(t,x,u) \end{aligned}$$

Analyze vs Discretize

There is a PMP for infinite horizon OCP and an HJB for finite horizon OCP but in the interests of time we shall not discuss them.

Analyze vs Discretize

There is a PMP for infinite horizon OCP and an HJB for finite horizon OCP but in the interests of time we shall not discuss them.

In order to solve these problems we have to discretize them. Discretize the optimal control problem or discretize the HJB or PMP equations?

Analyze vs Discretize

There is a PMP for infinite horizon OCP and an HJB for finite horizon OCP but in the interests of time we shall not discuss them.

In order to solve these problems we have to discretize them. Discretize the optimal control problem or discretize the HJB or PMP equations?

- **first analyze and then discretize**

Analyze vs Discretize

There is a PMP for infinite horizon OCP and an HJB for finite horizon OCP but in the interests of time we shall not discuss them.

In order to solve these problems we have to discretize them. Discretize the optimal control problem or discretize the HJB or PMP equations?

- **first analyze and then discretize**
- **first discretize and then analyze.**

Commutative Diagrams?

$$
\begin{array}{ccc}
\textbf{Infinite Horizon OCP} & \rightarrow & \textbf{Dynamic Program} \\
\downarrow & & \downarrow \\
\textbf{HJB PDE} & \rightarrow & \textbf{Dynamic Programming Equation}
\end{array}
$$

Commutative Diagrams?

$$\begin{array}{ccc}
\textbf{Infinite Horizon OCP} & \rightarrow & \textbf{Dynamic Program} \\
\downarrow & & \downarrow \\
\textbf{HJB PDE} & \rightarrow & \textbf{Dynamic Programming Equation}
\end{array}$$

$$\begin{array}{ccc}
\textbf{Finite Horizon OCP} & \rightarrow & \textbf{Nonlinear Program} \\
\downarrow & & \downarrow \\
\textbf{PMP} & \rightarrow & \textbf{Karush Kuhn Tucker Conditions}
\end{array}$$

Discretization of the HJB equation

For simplicty of exposition assume $n = 2,\ m = 1$. Choose a rectangle around $x = 0$ and partition it with stepsize h . Let $x^{i,j}$ denote the i, j node. Let $\pi^{i,j}$ be the current computed approximation to the optimal cost at the $x^{i,j}$.

Discretization of the HJB equation

For simplicty of exposition assume $n = 2,\ m = 1$. Choose a rectangle around $x = 0$ and partition it with stepsize h . Let $x^{i,j}$ denote the i, j node. Let $\pi^{i,j}$ be the current computed approximation to the optimal cost at the $x^{i,j}$.

For each i, j solve for the next approximation $\kappa^{i,j}$ to the optimal feedback

$$
\begin{aligned}
\kappa^{i,j} \;&=\; \operatorname{argmin}_u \Big\{ \left(\pi^{i+1,j} - \pi^{i-1,j}, \pi^{i,j+1} - \pi^{i,j-1}\right) f(x^{i,j}, u) \\
&\qquad +2hl(x^{i,j}, u) \Big\}
\end{aligned}
$$

Discretization of the HJB equation

For simplicty of exposition assume $n = 2,\ m = 1$. Choose a rectangle around $x = 0$ and partition it with stepsize h . Let $x^{i,j}$ denote the i, j node. Let $\pi^{i,j}$ be the current computed approximation to the optimal cost at the $x^{i,j}$.

For each i, j solve for the next approximation $\kappa^{i,j}$ to the optimal feedback

$$\begin{aligned}\kappa^{i,j} \;&=\; \text{argmin}_u \Big\{ \left(\pi^{i+1,j} - \pi^{i-1,j}, \pi^{i,j+1} - \pi^{i,j-1}\right) f(x^{i,j}, u) \\ &\qquad +2hl(x^{i,j}, u) \Big\}\end{aligned}$$

The next approximation to the optimal cost $\bar{\pi}^{i,j}$ is the solution to

$$\left(\bar{\pi}^{i+1,j} - \bar{\pi}^{i-1,j}, \bar{\pi}^{i,j+1} - \bar{\pi}^{i,j-1}\right) f(x^{i,j}\kappa^{i,j}) = -2hl(x^{i,j}, \kappa^{i,j})$$

The boundary condition is $\bar{\pi}^{i_0,j_0} = 0$ where $x^{i_0,j_0} = 0$.

Discretization of the HJB equation

For simplicty of exposition assume $n = 2,\ m = 1$. Choose a rectangle around $x = 0$ and partition it with stepsize h . Let $x^{i,j}$ denote the i, j node. Let $\pi^{i,j}$ be the current computed approximation to the optimal cost at the $x^{i,j}$.

For each i, j solve for the next approximation $\kappa^{i,j}$ to the optimal feedback

$$\begin{aligned}\kappa^{i,j} \quad &= \quad \text{argmin}_u \Big\{ \Big(\pi^{i+1,j} - \pi^{i-1,j}, \pi^{i,j+1} - \pi^{i,j-1} \Big) f(x^{i,j}, u) \\ &\qquad + 2hl(x^{i,j}, u) \Big\}\end{aligned}$$

The next approximation to the optimal cost $\bar{\pi}^{i,j}$ is the solution to

$$\Big(\bar{\pi}^{i+1,j} - \bar{\pi}^{i-1,j}, \bar{\pi}^{i,j+1} - \bar{\pi}^{i,j-1} \Big) f(x^{i,j} \kappa^{i,j}) = -2hl(x^{i,j}, \kappa^{i,j})$$

The boundary condition is $\bar{\pi}^{i_0,j_0} = 0$ where $x^{i_0,j_0} = 0$.

This is called policy iteration and it is not very efficient because it sweeps through the nodes many times.

Discretization of the Optimal Contol Problem

Instead we can discretize the optimal control problem. Asumme that the control takes on discrete values u_k and time is measured in steps of h.

Discretization of the Optimal Contol Problem

Instead we can discretize the optimal control problem. Asumme that the control takes on discrete values u_k and time is measured in steps of h.

Define $\bar{f}(x^{i,j}, u^k)$ to be the state node closest to

$$x^{i,j} + f(x^{i,j}, u^k)h$$

Then on the state and control grids we have the discrete dynamics

$$\begin{aligned} x^+ &= \bar{f}(x, u) \\ x(0) &= x^{i,j} \end{aligned}$$

Discretization of the Optimal Contol Problem

Instead we can discretize the optimal control problem. Asumme that the control takes on discrete values u_k and time is measured in steps of h.

Define $\bar{f}(x^{i,j}, u^k)$ to be the state node closest to

$$x^{i,j} + f(x^{i,j}, u^k)h$$

Then on the state and control grids we have the discrete dynamics

$$\begin{aligned} x^+ &= \bar{f}(x, u) \\ x(0) &= x^{i,j} \end{aligned}$$

and we minimize by choice of control sequence

$$\begin{aligned} \pi^{i,j} &= \min_{u(0:\infty)} \sum_{t=0:h:\infty} l(x, u)h \\ \kappa^{i,j} &= u^*(0) \end{aligned}$$

Discretization of the Optimal Contol Problem

Dynamic Programming Equation (DPE)

$$
\begin{aligned}
\pi^{i,j} &= \mathbf{min}_{u^k} \left\{ l(x^{i,j}, u^k)h + \pi(\bar{f}(x^{i,j}, u^k)) \right\} \\
\kappa^{i,j} &= \mathbf{argmin}_{u^k} \left\{ l(x^{i,j}, u^k)h + \pi(\bar{f}(x^{i,j}, u^k)) \right\}
\end{aligned}
$$

Discretization of the Optimal Contol Problem

Dynamic Programming Equation (DPE)

$$\begin{aligned}
\pi^{i,j} &= \min_{u^k} \left\{ l(x^{i,j}, u^k)h + \pi(\bar{f}(x^{i,j}, u^k)) \right\} \\
\kappa^{i,j} &= \operatorname{argmin}_{u^k} \left\{ l(x^{i,j}, u^k)h + \pi(\bar{f}(x^{i,j}, u^k)) \right\}
\end{aligned}$$

This can be solved by policy iteration. Given the current approximation $\pi(x^{i,j})$ to the optimal cost at grid points $x^{i,j}$, define the next approximation to the optimal feedback as

$$\kappa(x^{i,j}) = \operatorname{argmin}_{u^k} \left\{ l(x^{i,j}, u^k)h + \pi(\bar{f}(x^{i,j}, u^k)) \right\}$$

Discretization of the Optimal Contol Problem

Dynamic Programming Equation (DPE)

$$\begin{aligned}
\pi^{i,j} &= \min_{u^k} \left\{ l(x^{i,j}, u^k)h + \pi(\bar{f}(x^{i,j}, u^k)) \right\} \\
\kappa^{i,j} &= \operatorname{argmin}_{u^k} \left\{ l(x^{i,j}, u^k)h + \pi(\bar{f}(x^{i,j}, u^k)) \right\}
\end{aligned}$$

This can be solved by policy iteration. Given the current approximation $\pi(x^{i,j})$ to the optimal cost at grid points $x^{i,j}$, define the next approximation to the optimal feedback as

$$\kappa(x^{i,j}) = \operatorname{argmin}_{u^k} \left\{ l(x^{i,j}, u^k)h + \pi(\bar{f}(x^{i,j}, u^k)) \right\}$$

Given $\kappa(x^{i,j})$ then the next approximation to the optimal cost $\bar{\pi}(x^{i,j})$ is the solution of the equations

$$\bar{\pi}(x^{i,j}) = l(x^{i,j}, \kappa(x^{i,j}))h + \bar{\pi}(\bar{f}(x^{i,j}, \kappa(x^{i,j}))$$

Discretization of the Optimal Contol Problem

Dynamic Programming Equation (DPE)

$$
\begin{aligned}
\pi^{i,j} &= \min_{u^k}\left\{l(x^{i,j},u^k)h + \pi(\bar f(x^{i,j},u^k))\right\} \\
\kappa^{i,j} &= \operatorname{argmin}_{u^k}\left\{l(x^{i,j},u^k)h + \pi(\bar f(x^{i,j},u^k))\right\}
\end{aligned}
$$

This can be solved by policy iteration. Given the current approximation $\pi(x^{i,j})$ to the optimal cost at grid points $x^{i,j}$, define the next approximation to the optimal feedback as

$$
\kappa(x^{i,j}) = \operatorname{argmin}_{u^k}\left\{l(x^{i,j},u^k)h + \pi(\bar f(x^{i,j},u^k))\right\}
$$

Given $\kappa(x^{i,j})$ then the next approximation to the optimal cost $\bar\pi(x^{i,j})$ is the solution of the equations

$$
\bar\pi(x^{i,j}) = l(x^{i,j},\kappa(x^{i,j}))h + \bar\pi(\bar f(x^{i,j},\kappa(x^{i,j})))
$$

This is again policy iteration but it is still slow.

Approximation by a Markov Chain

Here is a simplistic version of this method.

Approximation by a Markov Chain

Here is a simplistic version of this method.

Partition the state space into a grid with spacing h and partition time with spacing k . Construct a Controlled Markov Chain with transition probability $p(x^1|x^0,u)$ from gridpoint x^0 to grid point x^1 with control u. Choose a search radius r and define

$$p(x^1|x^0,u) = \frac{\exp(-(\|x^1 - x^0 - (f(x^0,u)k\|^2)}{\rho(x^0,u)}$$

$$\rho(x^0,u) = \sum_j \exp(-(\|x^1 - x^0 - (f(x^0,u)k\|^2)$$

where the sum is over all gridpoints x^j such that

$$\|x^j - x^0 - (f(x^0,u)k\| \leq r$$

Approximation by a Markov Chain

Here is a simplistic version of this method.

Partition the state space into a grid with spacing h and partition time with spacing k . Construct a Controlled Markov Chain with transition probability $p(x^1|x^0,u)$ from gridpoint x^0 to grid point x^1 with control u. Choose a search radius r and define

$$\begin{aligned} p(x^1|x^0,u) &= \frac{\exp(-(\|x^1 - x^0 - (f(x^0,u)k\|^2)}{\rho(x^0,u)} \\ \rho(x^0,u) &= \sum_j \exp(-(\|x^1 - x^0 - (f(x^0,u)k\|^2) \end{aligned}$$

where the sum is over all gridpoints x^j such that

$$\|x^j - x^0 - (f(x^0,u)k\| \leq r$$

The cost is defined to be the expected value of

$$\sum_{t=0}^{\infty} l(x,u)$$

Approximation by a Markov Chain

The problem is solved via stochastic dynamic programming.

Approximation by a Markov Chain

The problem is solved via stochastic dynamic programming.

In dimensions $n = 2, 3$ and $m = 1, 2$ this is a feasible method.

Boue and Dupuis have proven that a more sophisticated version converges to the true solution as h, k go to 0 .

Approximation by a Markov Chain

The problem is solved via stochastic dynamic programming.

In dimensions $n = 2, 3$ and $m = 1, 2$ this is a feasible method.

Boue and Dupuis have proven that a more sophisticated version converges to the true solution as h, k go to 0 .

But in higher dimensions it difficult to implement.

Eikonal Equation

Suppose that the speed of propagation $c(x) > 0$ through a medium varies with location. Consider any path $x(t)$ between the source $x^0 = 0$ and x^1. Then the propagation time along this path is $\int_0^t 1\ d\tau$ so the Lagrangian $l(x,u) = 1$.

Eikonal Equation

Suppose that the speed of propagation $c(x) > 0$ through a medium varies with location. Consider any path $x(t)$ between the source $x^0 = 0$ and x^1. Then the propagation time along this path is $\int_0^t 1 \, d\tau$ so the Lagrangian $l(x, u) = 1$.

The dynamics is $\dot{x} = f(x, u) = c(x)u$ where $\|u\| = 1$.

Eikonal Equation

Suppose that the speed of propagation $c(x) > 0$ through a medium varies with location. Consider any path $x(t)$ between the source $x^0 = 0$ and x^1. Then the propagation time along this path is $\int_0^t 1 \, d\tau$ so the Lagrangian $l(x,u) = 1$.

The dynamics is $\dot{x} = f(x,u) = c(x)u$ where $\|u\| = 1$.

The HJB equations are

$$
\begin{aligned}
0 &= \min_{\|u\|=1} \left\{ \frac{\partial \pi}{\partial x}(x) c(x) u + 1 \right\} \\
\kappa(x) &= \operatorname{argmin}_{\|u\|=1} \left\{ \frac{\partial \pi}{\partial x}(x) c(x) u + 1 \right\}
\end{aligned}
$$

Eikonal Equation

Suppose that the speed of propagation $c(x) > 0$ through a medium varies with location. Consider any path $x(t)$ between the source $x^0 = 0$ and x^1. Then the propagation time along this path is $\int_0^t 1 \, d\tau$ so the Lagrangian $l(x, u) = 1$.

The dynamics is $\dot{x} = f(x, u) = c(x)u$ where $\|u\| = 1$.

The HJB equations are

$$\begin{aligned} 0 &= \min_{\|u\|=1} \left\{ \frac{\partial \pi}{\partial x}(x) c(x) u + 1 \right\} \\ \kappa(x) &= \operatorname{argmin}_{\|u\|=1} \left\{ \frac{\partial \pi}{\partial x}(x) c(x) u + 1 \right\} \end{aligned}$$

which reduce to

$$\left\| \frac{\partial \pi}{\partial x}(x) \right\| = \frac{1}{c(x)}, \qquad \kappa(x) = -\frac{\frac{\partial \pi}{\partial x}(x)}{\left\| \frac{\partial \pi}{\partial x}(x) \right\|}$$

Eikonal Equation

Suppose that the speed of propagation $c(x) > 0$ through a medium varies with location. Consider any path $x(t)$ between the source $x^0 = 0$ and x^1. Then the propagation time along this path is $\int_0^t 1 \, d\tau$ so the Lagrangian $l(x,u) = 1$.

The dynamics is $\dot{x} = f(x,u) = c(x)u$ where $\|u\| = 1$.

The HJB equations are

$$
\begin{aligned}
0 &= \min_{\|u\|=1} \left\{ \frac{\partial \pi}{\partial x}(x) c(x) u + 1 \right\} \\
\kappa(x) &= \operatorname{argmin}_{\|u\|=1} \left\{ \frac{\partial \pi}{\partial x}(x) c(x) u + 1 \right\}
\end{aligned}
$$

which reduce to

$$
\left\| \frac{\partial \pi}{\partial x}(x) \right\| = \frac{1}{c(x)}, \qquad \kappa(x) = -\frac{\frac{\partial \pi}{\partial x}(x)}{\left\| \frac{\partial \pi}{\partial x}(x) \right\|}
$$

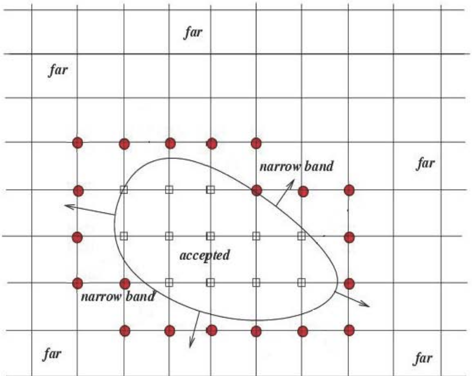
Fast Marching Method for the Eikonal Equation

This method is due to Tsitsiklis and it was refined by Sethian, Falcone and others.

Fast Marching Method for the Eikonal Equation

This method is due to Tsitsiklis and it was refined by Sethian, Falcone and others.

Partitition the nodes into three families called accepted, narrow band and far. Initially the only node in accepted family is the origin and $\pi^{i_0,j_0} = 0$.



Fast Marching Method for the Eikonal Equation

Assume that $\pi^{i,j}$ has been computed for all the accepted nodes.

For each node $x^{i,j}$ in the narrow band compute the rectilinear path to a node $x^{r,s}$ in the accepted region that minimizes the sum of travel time along the path plus $\pi^{r,s}$.

Fast Marching Method for the Eikonal Equation

Assume that $\pi^{i,j}$ has been computed for all the accepted nodes.

For each node $x^{i,j}$ in the narrow band compute the rectilinear path to a node $x^{r,s}$ in the accepted region that minimizes the sum of travel time along the path plus $\pi^{r,s}$.

This is typically done using Dykstra's method for finding the shortest path on a graph.

Fast Marching Method for the Eikonal Equation

Assume that $\pi^{i,j}$ has been computed for all the accepted nodes.

For each node $x^{i,j}$ in the narrow band compute the rectilinear path to a node $x^{r,s}$ in the accepted region that minimizes the sum of travel time along the path plus $\pi^{r,s}$.

This is typically done using Dykstra's method for finding the shortest path on a graph.

Then accept the narrow band node that minimizes this sum.

Fast Marching Method for the Eikonal Equation

Assume that $\pi^{i,j}$ has been computed for all the accepted nodes.

For each node $x^{i,j}$ in the narrow band compute the rectilinear path to a node $x^{r,s}$ in the accepted region that minimizes the sum of travel time along the path plus $\pi^{r,s}$.

This is typically done using Dykstra's method for finding the shortest path on a graph.

Then accept the narrow band node that minimizes this sum.

Repeat until the accepted nodes cover the region where the solution is desired.

Fast Marching Method for the Eikonal Equation

Assume that $\pi^{i,j}$ has been computed for all the accepted nodes.

For each node $x^{i,j}$ in the narrow band compute the rectilinear path to a node $x^{r,s}$ in the accepted region that minimizes the sum of travel time along the path plus $\pi^{r,s}$.

This is typically done using Dykstra's method for finding the shortest path on a graph.

Then accept the narrow band node that minimizes this sum.

Repeat until the accepted nodes cover the region where the solution is desired.

Different implimentations of FMM use different ways of computing the sum of travel time along the path plus $\pi^{r,s}$.

Fast Marching Method for the Eikonal Equation

Assume that $\pi^{i,j}$ has been computed for all the accepted nodes.

For each node $x^{i,j}$ in the narrow band compute the rectilinear path to a node $x^{r,s}$ in the accepted region that minimizes the sum of travel time along the path plus $\pi^{r,s}$.

This is typically done using Dykstra's method for finding the shortest path on a graph.

Then accept the narrow band node that minimizes this sum.

Repeat until the accepted nodes cover the region where the solution is desired.

Different implimentations of FMM use different ways of computing the sum of travel time along the path plus $\pi^{r,s}$.

The advantage of FMM is that the computation visits each node much less often than in the naive approach.

Fast Marching Method for the Eikonal Equation

Assume that $\pi^{i,j}$ has been computed for all the accepted nodes.

For each node $x^{i,j}$ in the narrow band compute the rectilinear path to a node $x^{r,s}$ in the accepted region that minimizes the sum of travel time along the path plus $\pi^{r,s}$.

This is typically done using Dykstra's method for finding the shortest path on a graph.

Then accept the narrow band node that minimizes this sum.

Repeat until the accepted nodes cover the region where the solution is desired.

Different implimentations of FMM use different ways of computing the sum of travel time along the path plus $\pi^{r,s}$.

The advantage of FMM is that the computation visits each node much less often than in the naive approach.

The FFM has been generalized to other optimal control problems but computing the minimum sums is more complicated because not every rectilinear path is feasible.

Curse of Dimensionality

All methods for solving HJB equations suffer from Bellman's

Curse of Dimensionality

Curse of Dimensionality

All methods for solving HJB equations suffer from Bellman's

Curse of Dimensionality

Practical optimal control problems usually have state dimension larger than 2 or 3. For example, the attitude control problem for a spacecraft has state dimension $n = 6$ and control dimension typically $m = 3$. The position and attitude control problem for an airplane has state dimension $n = 12$ and control dimension at least $m = 4$.

Curse of Dimensionality

All methods for solving HJB equations suffer from Bellman's

Curse of Dimensionality

Practical optimal control problems usually have state dimension larger than 2 or 3. For example, the attitude control problem for a spacecraft has state dimension $n = 6$ and control dimension typically $m = 3$. The position and attitude control problem for an airplane has state dimension $n = 12$ and control dimension at least $m = 4$.

Consider trying to apply a grid based method.. For the solution to be reasonably accurate we would need a substantial number of grid points in each coordinate direction, e.g., 10^2. Then the total number of grid points is 10^{12} for attitude control and 10^{24} for position and attitude control. If we can process 100 nodes a second that works out to about 300 years for attitude control and $3 \cdot 10^{14}$ years for position and attitude control.

HJB Equations and Conservation Laws

Suppose we have a time varying problem of the form

$$\begin{aligned}\dot{x} &= f(t,x) + g(t,x)u \\ l(t,x,u) &= q(t,x) + \frac{1}{2}u'R(t,x)u\end{aligned}$$

Then the HJB PDEs reduce to the HJ PDE

$$0 = \frac{\partial \pi}{\partial t} + \frac{\partial \pi}{\partial x}f - \frac{1}{2}\frac{\partial \pi}{\partial x}gRg'\left(\frac{\partial \pi}{\partial x}\right)' + q$$

Let $p = \frac{\partial \pi}{\partial x}$ and take the Jacobian of the HJ equation to obtain the conservation law

$$0 = \frac{\partial p}{\partial t} + \frac{\partial}{\partial x}F(t,x,p)$$

where the flux term is

$$F(t,x,p) = pf - \frac{1}{2}pgRg'p' + q$$

HJB Equations and Conservation Laws

This connection has been used to take advantage of the highly developed methods for conservation laws to solve HJ and HJB equations.

HJB Equations and Conservation Laws

This connection has been used to take advantage of the highly developed methods for conservation laws to solve HJ and HJB equations.

But there are difficulties, one is that we have increased the dimension of the unkown.

$$\pi(x) \in I\!R, \qquad p(x) \in I\!R^{1\times n}$$

HJB Equations and Conservation Laws

This connection has been used to take advantage of the highly developed methods for conservation laws to solve HJ and HJB equations.

But there are difficulties, one is that we have increased the dimension of the unkown.

$$\pi(x) \in I\!R, \qquad p(x) \in I\!R^{1\times n}$$

Another is that we are looking for a solution of the conservation law that is a closed one form,

$$\frac{\partial p_i}{\partial x_j} = \frac{\partial p_j}{\partial x_i}$$

Invariant Manifold Methods

Hamilton Differential Equations

$$\dot{x}_i = \frac{\partial \mathcal{H}}{\partial p_i}, \qquad \dot{p}_i = -\frac{\partial \mathcal{H}}{\partial x_i}$$

Invariant Manifold Methods

Hamilton Differential Equations

$$\dot{x}_i = \frac{\partial \mathcal{H}}{\partial p_i}, \qquad \dot{p}_i = -\frac{\partial \mathcal{H}}{\partial x_i}$$

Under suitable conditions the linear part of these equations have n eigenvalues in the open left half plane and n eigenvalues in the open right half plane so there is an n dimensional stable manifold.

Invariant Manifold Methods

Hamilton Differential Equations

$$\dot{x}_i = \frac{\partial \mathcal{H}}{\partial p_i}, \qquad \dot{p}_i = -\frac{\partial \mathcal{H}}{\partial x_i}$$

Under suitable conditions the linear part of these equations have n eigenvalues in the open left half plane and n eigenvalues in the open right half plane so there is an n dimensional stable manifold.

This stable manifold is the graph of a mapping

$$x \;\mapsto\; p(x)$$

Invariant Manifold Methods

Hamilton Differential Equations

$$\dot{x}_i = \frac{\partial \mathcal{H}}{\partial p_i}, \qquad \dot{p}_i = -\frac{\partial \mathcal{H}}{\partial x_i}$$

Under suitable conditions the linear part of these equations have n eigenvalues in the open left half plane and n eigenvalues in the open right half plane so there is an n dimensional stable manifold.

This stable manifold is the graph of a mapping

$$x \ \mapsto \ p(x)$$

Because of the Hamiltonian structure this manifold is Lagrangian, i.e., a gradient and

$$p(x) \ = \ \frac{\partial \pi}{\partial x}(x)$$

Invariant Manifold Methods

Hamilton Differential Equations

$$\dot{x}_i = \frac{\partial \mathcal{H}}{\partial p_i}, \qquad \dot{p}_i = -\frac{\partial \mathcal{H}}{\partial x_i}$$

Under suitable conditions the linear part of these equations have n eigenvalues in the open left half plane and n eigenvalues in the open right half plane so there is an n dimensional stable manifold.

This stable manifold is the graph of a mapping

$$x \ \mapsto \ p(x)$$

Because of the Hamiltonian structure this manifold is Lagrangian, i.e., a gradient and

$$p(x) \ = \ \frac{\partial \pi}{\partial x}(x)$$

So we can compute $\pi(x)$ by computing the stable manifold of the Hamiltonian dynamics.

Hauser-Osinga Method

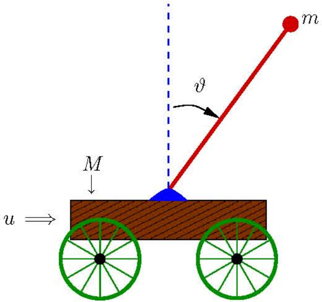


Figure 1: *Sketch of the balanced planar pendulum on a moving cart.*

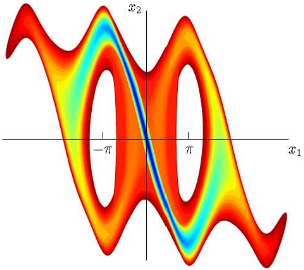


Figure 2: *Each point is colored according to how high the cost of getting to the origin using this point as initial condition. The cost increases as the color changes from blue to red.*

Min-Plus Methods

The min-plus semiring is defined as follows

$$\begin{aligned} \boldsymbol{\xi} \oplus \boldsymbol{\zeta} &= \mathbf{min}\{\boldsymbol{\xi}, \boldsymbol{\zeta}\} \\ \boldsymbol{\xi} \otimes \boldsymbol{\zeta} &= \boldsymbol{\xi} + \boldsymbol{\zeta} \end{aligned}$$

Min-Plus Methods

The min-plus semiring is defined as follows

$$\begin{aligned} \xi \oplus \zeta &= \min\{\xi, \zeta\} \\ \xi \otimes \zeta &= \xi + \zeta \end{aligned}$$

These operations are commutative, associative and distributive,

$$\xi \otimes (\zeta \oplus \eta) = (\xi \otimes \zeta) \oplus (\xi \otimes \eta)$$

The $\oplus$ "identity" is ∞ , the $\otimes$ "identity" is $\mathbf{0}$.

Min-Plus Methods

The min-plus semiring is defined as follows

$$\begin{aligned} \xi \oplus \zeta &= \min\{\xi, \zeta\} \\ \xi \otimes \zeta &= \xi + \zeta \end{aligned}$$

These operations are commutative, associative and distributive,

$$\xi \otimes (\zeta \oplus \eta) = (\xi \otimes \zeta) \oplus (\xi \otimes \eta)$$

The $\oplus$ "identity" is ∞ , the $\otimes$ "identity" is $\mathbf{0}$.

It is not an algebra because there are no $\oplus$ inverses. We cannot solve for ζ the equation

$$\xi \oplus \zeta = \min\{\xi, \zeta\} = \infty$$

Min-Plus Methods

The min-plus semiring is defined as follows

$$\begin{aligned} \xi \oplus \zeta &= \min\{\xi, \zeta\} \\ \xi \otimes \zeta &= \xi + \zeta \end{aligned}$$

These operations are commutative, associative and distributive,

$$\xi \otimes (\zeta \oplus \eta) = (\xi \otimes \zeta) \oplus (\xi \otimes \eta)$$

The $\oplus$ "identity" is ∞ , the $\otimes$ "identity" is 0.

It is not an algebra because there are no $\oplus$ inverses. We cannot solve for ζ the equation

$$\xi \oplus \zeta = \min\{\xi, \zeta\} = \infty$$

There is also a max-plus semiring where

$$\xi \oplus \zeta = \max\{\xi, \zeta\}$$

Min-Plus Methods

Consider the semigroup acting on function $\psi(x)$ for $T \geq 0$

$$\begin{aligned}
\mathcal{S}_T(\phi)(x^T) &= \min_{u(-T:0)} \left\{ \int_{-T}^{0} l(x(t), u(t))\ dt + \phi(x(0)) \right\} \\
\dot{x} &= f(x, u) \\
x(-T) &= x^T
\end{aligned}$$

Min-Plus Methods

Consider the semigroup acting on function $\psi(x)$ for $T \geq 0$

$$
\begin{aligned}
\mathcal{S}_T(\phi)(x^T) &= \min_{u(-T:0)} \left\{ \int_{-T}^{0} l(x(t), u(t))\ dt + \phi(x(0)) \right\} \\
\dot{x} &= f(x, u) \\
x(-T) &= x^T
\end{aligned}
$$

This semigroup is nonlinear in the ordinary sense and the optimal cost $\pi(x)$ is a fixed point of this semigroup,

$$
\pi(x) = \mathcal{S}_T(\pi)(x)
$$

Min-Plus Methods

Consider the semigroup acting on function $\psi(x)$ for $T \geq 0$

$$\begin{aligned}
\mathcal{S}_T(\phi)(x^T) &= \min_{u(-T:0)} \left\{ \int_{-T}^{0} l(x(t), u(t))\ dt + \phi(x(0)) \right\} \\
\dot{x} &= f(x, u) \\
x(-T) &= x^T
\end{aligned}$$

This semigroup is nonlinear in the ordinary sense and the optimal cost $\pi(x)$ is a fixed point of this semigroup,

$$\pi(x) = \mathcal{S}_T(\pi)(x)$$

But is linear in the min-plus sense and $\pi(x)$ is an eigenvector corresponding to the eigenvalue $\mathbf{0}$ which is the $\otimes$ identity.

$$0 \otimes \pi(x) = \mathcal{S}_T(\pi(x)$$

Min-Plus Methods

The power method is the standard way to find an eigenvector.

Min-Plus Methods

The power method is the standard way to find an eigenvector.

Start with any $\phi(x)$ and compute

$$\pi(x) = \lim_{N \to \infty} \mathcal{S}_T(\cdot) \circ \mathcal{S}_T(\cdot) \circ \cdots \circ \mathcal{S}_T(\phi)(x)$$

where N is the number of composition factors.

Min-Plus Methods

The power method is the standard way to find an eigenvector.

Start with any $\phi(x)$ and compute

$$\pi(x) \;=\; \lim_{N\to\infty} \mathcal{S}_T(\cdot)\circ\mathcal{S}_T(\cdot)\circ\cdots\circ\mathcal{S}_T(\phi)(x)$$

where N is the number of composition factors.

To make the calculation finite dimensional $\pi(x)$ is chosen as a min-plus combination of basis functions.

$$\pi(x) \;=\; (\alpha_1\otimes\psi_1(x))\oplus\cdots\oplus(\alpha_k\otimes\psi_k(x))$$

and a projection is done after each application of the semigroup.

Min-Plus Methods

The power method is the standard way to find an eigenvector.

Start with any $\phi(x)$ and compute

$$\pi(x) \;=\; \lim_{N\to\infty} \mathcal{S}_T(\cdot)\circ\mathcal{S}_T(\cdot)\circ\cdots\circ\mathcal{S}_T(\phi)(x)$$

where N is the number of composition factors.

To make the calculation finite dimensional $\pi(x)$ is chosen as a min-plus combination of basis functions.

$$\pi(x) \;=\; (\alpha_1\otimes\psi_1(x))\oplus\cdots\oplus(\alpha_k\otimes\psi_k(x))$$

and a projection is done after each application of the semigroup.

This is very similar to policy iteration, the princple difference is the restriction to min-plus combinations of basis functions.

Min-Plus Methods

The power method is the standard way to find an eigenvector.

Start with any $\phi(x)$ and compute

$$\pi(x) \;=\; \lim_{N\to\infty} \mathcal{S}_T(\cdot)\circ\mathcal{S}_T(\cdot)\circ\cdots\circ\mathcal{S}_T(\phi)(x)$$

where N is the number of composition factors.

To make the calculation finite dimensional $\pi(x)$ is chosen as a min-plus combination of basis functions.

$$\pi(x) \;=\; (\alpha_1\otimes\psi_1(x))\oplus\cdots\oplus(\alpha_k\otimes\psi_k(x))$$

and a projection is done after each application of the semigroup.

This is very similar to policy iteration, the princple difference is the restriction to min-plus combinations of basis functions.

The number of basis functions needed for a given accuracy is exponential in the state dimension n but it is probably grows slower than the number of grid points.

Min-Plus Methods

McEneaney has developed a method for solving HJB equations for Hamiltonians that are the minimum (or maxima) of a family of Hamiltonian of LQR problems.

Min-Plus Methods

McEneaney has developed a method for solving HJB equations for Hamiltonians that are the minimum (or maxima) of a family of Hamiltonian of LQR problems.

It is too complicated to discuss here but it uses min-plus techniques and duality.

Min-Plus Methods

McEneaney has developed a method for solving HJB equations for Hamiltonians that are the minimum (or maxima) of a family of Hamiltonian of LQR problems.

It is too complicated to discuss here but it uses min-plus techniques and duality.

No grid is necessary so it does not suffer from the curse of dimensionality on that score.

Min-Plus Methods

McEneaney has developed a method for solving HJB equations for Hamiltonians that are the minimum (or maxima) of a family of Hamiltonian of LQR problems.

It is too complicated to discuss here but it uses min-plus techniques and duality.

No grid is necessary so it does not suffer from the curse of dimensionality on that score.

But the computational cost of the method grows exponentialy in the number of LQR Hamiltonians. The number of LQR Hamiltonians needed probably increases with the state dimension but more slowy than the number of gridpoints does.

Min-Plus Methods

McEneaney has developed a method for solving HJB equations for Hamiltonians that are the minimum (or maxima) of a family of Hamiltonian of LQR problems.

It is too complicated to discuss here but it uses min-plus techniques and duality.

No grid is necessary so it does not suffer from the curse of dimensionality on that score.

But the computational cost of the method grows exponentialy in the number of LQR Hamiltonians. The number of LQR Hamiltonians needed probably increases with the state dimension but more slowy than the number of gridpoints does.

It also suffers from a curse of complexity as it requires computing the pointwise maxima (or minima) of a large number of functions which can be expensive.

Higher Order Methods

Why go to higher order methods? Here is a simple answer.

Higher Order Methods

Why go to higher order methods? Here is a simple answer.

If the true solution is smooth enough then the local truncation error of a first order method with step size h_1 is $O(h_1)^2$.

Higher Order Methods

Why go to higher order methods? Here is a simple answer.

If the true solution is smooth enough then the local truncation error of a first order method with step size h_1 is $O(h_1)^2$.

If the true solution is smooth enough then the local truncation error of a third order method with step size h_3 is $O(h_3)^4$.

Higher Order Methods

Why go to higher order methods? Here is a simple answer.

If the true solution is smooth enough then the local truncation error of a first order method with step size h_1 is $O(h_1)^2$.

If the true solution is smooth enough then the local truncation error of a third order method with step size h_3 is $O(h_3)^4$.

Assuming the order constants are about the same size then to achieve the same accuracy $h_1^2 \approx h_3^4$ or $h_3 \approx \sqrt{h_1}$.

Higher Order Methods

Why go to higher order methods? Here is a simple answer.

If the true solution is smooth enough then the local truncation error of a first order method with step size h_1 is $O(h_1)^2$.

If the true solution is smooth enough then the local truncation error of a third order method with step size h_3 is $O(h_3)^4$.

Assuming the order constants are about the same size then to achieve the same accuracy $h_1^2 \approx h_3^4$ or $h_3 \approx \sqrt{h_1}$.

If $h_1 = 0.01$ then $h_3 \approx 0.1$ so the number of grid points that is needed for a given accuracy is reduced by a factor of 10 in each dimension. If the state dimension is n then the reduction in grid points is by a factor of 10^n .

Higher Order Methods

Why go to higher order methods? Here is a simple answer.

If the true solution is smooth enough then the local truncation error of a first order method with step size h_1 is $O(h_1)^2$.

If the true solution is smooth enough then the local truncation error of a third order method with step size h_3 is $O(h_3)^4$.

Assuming the order constants are about the same size then to achieve the same accuracy $h_1^2 \approx h_3^4$ or $h_3 \approx \sqrt{h_1}$.

If $h_1 = 0.01$ then $h_3 \approx 0.1$ so the number of grid points that is needed for a given accuracy is reduced by a factor of 10 in each dimension. If the state dimension is n then the reduction in grid points is by a factor of 10^n .

If the third order method takes $k_3(n)$ times longer to compute for each node then the reduction in computational time is by the factor $\frac{10^n}{k_3(n)}$. Typically $k_3(n)$ is polynomial in n.

Higher Order Methods

Of course to use a higher order method the solution must be smooth enough.

Higher Order Methods

Of course to use a higher order method the solution must be smooth enough.

The computations become more difficult and less accurate.

Higher Order Methods

Of course to use a higher order method the solution must be smooth enough.

The computations become more difficult and less accurate.

There are diminishing returns as we go to higher orders. Consider a fifth order method with step size h_5 . Then for same level of accuracy

$$h_1^2 \approx h_5^6 \quad \text{so} \quad h_5 \approx h_1^{1/3}$$
$$h_5 \approx 0.2154 \quad \text{when} \quad h_1 = 0.01$$

Higher Order Methods

Of course to use a higher order method the solution must be smooth enough.

The computations become more difficult and less accurate.

There are diminishing returns as we go to higher orders. Consider a fifth order method with step size h_5 . Then for same level of accuracy

$$h_1^2 \approx h_5^6 \qquad \text{so} \qquad h_5 \approx h_1^{1/3}$$
$$h_5 \approx 0.2154 \quad \text{when} \quad h_1 = 0.01$$

So a third order method requires a factor of about 2^n more grid points than a fifth order method.

Higher Order Methods

Of course to use a higher order method the solution must be smooth enough.

The computations become more difficult and less accurate.

There are diminishing returns as we go to higher orders. Consider a fifth order method with step size h_5 . Then for same level of accuracy

$$h_1^2 \approx h_5^6 \qquad \text{so} \qquad h_5 \approx h_1^{1/3}$$
$$h_5 \approx 0.2154 \quad \text{when} \quad h_1 = 0.01$$

So a third order method requires a factor of about 2^n more grid points than a fifth order method.

Recall that a first order method requires a factor of about 10^n more grid points than a third order method.

Higher Order Methods

Of course to use a higher order method the solution must be smooth enough.

The computations become more difficult and less accurate.

There are diminishing returns as we go to higher orders. Consider a fifth order method with step size h_5 . Then for same level of accuracy

$$h_1^2 \approx h_5^6 \quad \text{so} \quad h_5 \approx h_1^{1/3}$$
$$h_5 \approx 0.2154 \quad \text{when} \quad h_1 = 0.01$$

So a third order method requires a factor of about 2^n more grid points than a fifth order method.

Recall that a first order method requires a factor of about 10^n more grid points than a third order method.

Suppose the fifth order method takes $k_5(n)$ times longer for each node. Typically $k_d(n)$ grows exponentially in d.

Richardson Extrapolation

This is the simplest way of generating a higher order method.

Richardson Extrapolation

This is the simplest way of generating a higher order method.

Suppose we have a first order method $M_1(h)$ for solving a problem using stepsize h . If the problem and the method are smooth enough then we expect that the error is a power series in h with lowest order term a constant times h^2 . Let α denote the true solution then with steps sizes h and $2h$

$$\begin{aligned}
\alpha &= M_1(h) + \beta_2 h^2 + \beta_3 h^3 + \ldots \\
\alpha &= M_1(2h) + \beta_2 (2h)^2 + \beta_3 (2h)^3 + \ldots
\end{aligned}$$

Richardson Extrapolation

This is the simplest way of generating a higher order method.

Suppose we have a first order method $M_1(h)$ for solving a problem using stepsize h . If the problem and the method are smooth enough then we expect that the error is a power series in h with lowest order term a constant times h^2 . Let α denote the true solution then with steps sizes h and $2h$

$$\begin{aligned}\alpha &= M_1(h) + \beta_2 h^2 + \beta_3 h^3 + \ldots \\ \alpha &= M_1(2h) + \beta_2 (2h)^2 + \beta_3 (2h)^3 + \ldots\end{aligned}$$

Multiply the first by $4/3$ and the second by $-1/3$ and add,

$$\begin{aligned}\alpha &= M_2(h) + O(h)^3 \\ \text{where } M_2(h) &= \frac{4}{3}M_1(h) - \frac{1}{3}M_1(2h)\end{aligned}$$

Richardson Extrapolation

This is the simplest way of generating a higher order method.

Suppose we have a first order method $M_1(h)$ for solving a problem using stepsize h . If the problem and the method are smooth enough then we expect that the error is a power series in h with lowest order term a constant times h^2 . Let α denote the true solution then with steps sizes h and $2h$

$$\begin{aligned} \alpha &= M_1(h) + \beta_2 h^2 + \beta_3 h^3 + \ldots \\ \alpha &= M_1(2h) + \beta_2 (2h)^2 + \beta_3 (2h)^3 + \ldots \end{aligned}$$

Multiply the first by $4/3$ and the second by $-1/3$ and add,

$$\begin{aligned} \alpha &= M_2(h) + O(h)^3 \\ \text{where } M_2(h) &= \frac{4}{3} M_1(h) - \frac{1}{3} M_1(2h) \end{aligned}$$

Szpiro and Dupuis have applied this technique to HJB equations.

Singular PDEs

A first order quasilinear PDE

$$0 = \frac{\partial \phi}{\partial x}(x)a(x) + b(x, \phi(x))$$

is said to be singular at $x = 0$ if $a(0) = 0$. Usually $b(0,0) = 0$.

Singular PDEs

A first order quasilinear PDE

$$0 = \frac{\partial \phi}{\partial x}(x) a(x) + b(x, \phi(x))$$

is said to be singular at $x = 0$ if $a(0) = 0$. Usually $b(0,0) = 0$.

We expand in power series.

$$\begin{aligned} a(x) &= Ax + a^{[2]}(x) + a^{[3]}(x) + \ldots \\ b(x,\phi(x)) &= Cx + B\phi(x) + (b(x,\phi(x)))^{[2]} + (b(x,\phi(x)))^{[3]} + \ldots \\ \phi(x) &= Tx + \phi^{[2]}(x) + \phi^{[3]}(x) + \ldots \end{aligned}$$

where $(\cdot)^{[d]}$ denotes terms homogeneous of degree d.

Singular PDEs

Collect terms of first degree.

$$TA + BT = -C$$

Singular PDEs

Collect terms of first degree.

$$TA + BT = -C$$

This is solvable for any C iff there is no resonance of the form $\alpha_i + \beta_j = 0$ where α_i is an eigenvalue of A and β_j is an eigenvalue of B.

Singular PDEs

Collect terms of first degree.

$$TA + BT = -C$$

This is solvable for any C iff there is no resonance of the form $\alpha_i + \beta_j = 0$ where α_i is an eigenvalue of A and β_j is an eigenvalue of B.

Having found T we collect quadratic terms to get an equation of the form

$$\frac{\partial \phi^{[2]}}{\partial x}(x)Ax + B\phi^{[2]}(x) = \text{known terms}$$

Singular PDEs

Collect terms of first degree.

$$TA + BT \;=\; -C$$

This is solvable for any C iff there is no resonance of the form $\alpha_i + \beta_j = 0$ where α_i is an eigenvalue of A and β_j is an eigenvalue of B.

Having found T we collect quadratic terms to get an equation of the form

$$\frac{\partial \phi^{[2]}}{\partial x}(x)Ax + B\phi^{[2]}(x) \;=\; \text{known terms}$$

Notice the map

$$\phi^{[2]}(x) \;\mapsto\; \frac{\partial \phi^{[2]}}{\partial x}(x)Ax + B\phi^{[2]}(x)$$

takes quadratic polynomials to quadratic polynomials. It is invertible iff there is no resonance of the form

$$\alpha_{i_1} + \alpha_{i_2} + \beta_j = 0$$

Singular PDEs

Collect terms of first degree.

$$TA + BT = -C$$

This is solvable for any C iff there is no resonance of the form $\alpha_i + \beta_j = 0$ where α_i is an eigenvalue of A and β_j is an eigenvalue of B.

Having found T we collect quadratic terms to get an equation of the form

$$\frac{\partial \phi^{[2]}}{\partial x}(x)Ax + B\phi^{[2]}(x) = \text{known terms}$$

Notice the map

$$\phi^{[2]}(x) \mapsto \frac{\partial \phi^{[2]}}{\partial x}(x)Ax + B\phi^{[2]}(x)$$

takes quadratic polynomials to quadratic polynomials. It is invertible iff there is no resonance of the form

$$\alpha_{i_1} + \alpha_{i_2} + \beta_j = 0$$

The higher degrees terms can be found in a similar fashion.

Singular PDEs

Many of the most important PDEs of nonlinear dynamics and control are essentially singular first order quasilinear including

- **PDEs for Stable, Unstable, Center, etc. Manifolds**

Singular PDEs

Many of the most important PDEs of nonlinear dynamics and control are essentially singular first order quasilinear including

- **PDEs for Stable, Unstable, Center, etc. Manifolds**
- **Hamilton Jacobi Bellman PDEs**

Singular PDEs

Many of the most important PDEs of nonlinear dynamics and control are essentially singular first order quasilinear including

- **PDEs for Stable, Unstable, Center, etc. Manifolds**
- **Hamilton Jacobi Bellman PDEs**
- **Hamilton Jacobi Isaacs PDEs**

Singular PDEs

Many of the most important PDEs of nonlinear dynamics and control are essentially singular first order quasilinear including

- **PDEs for Stable, Unstable, Center, etc. Manifolds**
- **Hamilton Jacobi Bellman PDEs**
- **Hamilton Jacobi Isaacs PDEs**
- **Francis Byrnes Isidori PDE of Nonlinear Regulation**

Singular PDEs

Many of the most important PDEs of nonlinear dynamics and control are essentially singular first order quasilinear including

- **PDEs for Stable, Unstable, Center, etc. Manifolds**
- **Hamilton Jacobi Bellman PDEs**
- **Hamilton Jacobi Isaacs PDEs**
- **Francis Byrnes Isidori PDE of Nonlinear Regulation**
- **Kazantzis Kravaris PDE of Nonlinear Estimation**

Singular PDEs

Many of the most important PDEs of nonlinear dynamics and control are essentially singular first order quasilinear including

- **PDEs for Stable, Unstable, Center, etc. Manifolds**
- **Hamilton Jacobi Bellman PDEs**
- **Hamilton Jacobi Isaacs PDEs**
- **Francis Byrnes Isidori PDE of Nonlinear Regulation**
- **Kazantzis Kravaris PDE of Nonlinear Estimation**

Singular PDEs

Many of the most important PDEs of nonlinear dynamics and control are essentially singular first order quasilinear including

- **PDEs for Stable, Unstable, Center, etc. Manifolds**
- **Hamilton Jacobi Bellman PDEs**
- **Hamilton Jacobi Isaacs PDEs**
- **Francis Byrnes Isidori PDE of Nonlinear Regulation**
- **Kazantzis Kravaris PDE of Nonlinear Estimation**

Singular PDEs

Many of the most important PDEs of nonlinear dynamics and control are essentially singular first order quasilinear including

- **PDEs for Stable, Unstable, Center, etc. Manifolds**
- **Hamilton Jacobi Bellman PDEs**
- **Hamilton Jacobi Isaacs PDEs**
- **Francis Byrnes Isidori PDE of Nonlinear Regulation**
- **Kazantzis Kravaris PDE of Nonlinear Estimation**

Discrete time and time varying problems can also be solved by similar power series methods.

Singular PDEs

Many of the most important PDEs of nonlinear dynamics and control are essentially singular first order quasilinear including

- **PDEs for Stable, Unstable, Center, etc. Manifolds**
- **Hamilton Jacobi Bellman PDEs**
- **Hamilton Jacobi Isaacs PDEs**
- **Francis Byrnes Isidori PDE of Nonlinear Regulation**
- **Kazantzis Kravaris PDE of Nonlinear Estimation**

Discrete time and time varying problems can also be solved by similar power series methods.

In discrete time the degree two nonresonance conditions are

$$\alpha_{i_1}\alpha_{i_2} \neq \beta_j$$

Al'brecht's Method

Al'brecht developed the power series method for HJB equations for the optimal cost and optimal feedback,

$$\begin{aligned}\pi(x) &= \frac{1}{2}x'Px + \pi^{[3]}(x) + \pi^{[4]}(x) + \dots \\ \kappa(x) &= Kx + \kappa^{[2]}(x) + \kappa^{[3]}(x) + \dots\end{aligned}$$

Al'brecht's Method

Al'brecht developed the power series method for HJB equations for the optimal cost and optimal feedback,

$$\begin{aligned}\pi(x) &= \frac{1}{2}x'Px + \pi^{[3]}(x) + \pi^{[4]}(x) + \ldots \\ \kappa(x) &= Kx + \kappa^{[2]}(x) + \kappa^{[3]}(x) + \ldots\end{aligned}$$

At the lowest degree we get the familiar LQR equations

$$\begin{aligned}0 &= F'P + PF + Q - PGR^{-1}G'P \\ K &= -R^{-1}G'P\end{aligned}$$

Al'brecht's Method

Al'brecht developed the power series method for HJB equations for the optimal cost and optimal feedback,

$$\begin{aligned} \pi(x) &= \frac{1}{2}x'Px + \pi^{[3]}(x) + \pi^{[4]}(x) + \dots \\ \kappa(x) &= Kx + \kappa^{[2]}(x) + \kappa^{[3]}(x) + \dots \end{aligned}$$

At the lowest degree we get the familiar LQR equations

$$\begin{aligned} 0 &= F'P + PF + Q - PGR^{-1}G'P \\ K &= -R^{-1}G'P \end{aligned}$$

If the standard LQR conditions are satisfied then the Riccati equation has an unique nonnegative definite solution P and the linear part of the closed loop dynamics

$$\dot{x} = (F + GK)x$$

is exponentially stable.

Al'brecht's Method

Al'brecht developed the power series method for HJB equations for the optimal cost and optimal feedback,

$$\begin{aligned}\pi(x) &= \frac{1}{2}x'Px + \pi^{[3]}(x) + \pi^{[4]}(x) + \ldots \\ \kappa(x) &= Kx + \kappa^{[2]}(x) + \kappa^{[3]}(x) + \ldots\end{aligned}$$

At the lowest degree we get the familiar LQR equations

$$\begin{aligned}0 &= F'P + PF + Q - PGR^{-1}G'P \\ K &= -R^{-1}G'P\end{aligned}$$

If the standard LQR conditions are satisfied then the Riccati equation has an unique nonnegative definite solution P and the linear part of the closed loop dynamics

$$\dot{x} = (F + GK)x$$

is exponentially stable.

This guarantees there are no resonances so the higher degree terms of π, κ can be found by solving invertible linear equations.

Al'brecht's Method

This method has been implimented in the Nonlinear Systems Toolbox.

Al'brecht's Method

This method has been implimented in the Nonlinear Systems Toolbox.

The HJB equations can be solved to degree 4 in $\pi(x)$ and degree 3 in $\kappa(x)$ for systems with state dimension $n = 25$ and control dimension $m = 8$ on a lap top.

Pros and Cons of Power Series Methods

- **Power Series Methods can also be used for discrete time and time varing problems.**

Pros and Cons of Power Series Methods

- **Power Series Methods can also be used for discrete time and time varing problems.**
- **They involve mostly solving linear equations and Matlab software is available.**

Pros and Cons of Power Series Methods

- **Power Series Methods can also be used for discrete time and time varing problems.**
- **They involve mostly solving linear equations and Matlab software is available.**
- **They are restricted to smooth systems with no state or control constraints.**

Pros and Cons of Power Series Methods

- **Power Series Methods can also be used for discrete time and time varing problems.**
- **They involve mostly solving linear equations and Matlab software is available.**
- **They are restricted to smooth systems with no state or control constraints.**
- **The LQR part must yield a Hurwitz $F + GK$.**

Pros and Cons of Power Series Methods

- **Power Series Methods can also be used for discrete time and time varing problems.**
- **They involve mostly solving linear equations and Matlab software is available.**
- **They are restricted to smooth systems with no state or control constraints.**
- **The LQR part must yield a Hurwitz $F + GK$.**
- **The software can used for systems of moderately large state dimension, e.g., $n = 25$, $m = 8$.**

Pros and Cons of Power Series Methods

- **Power Series Methods can also be used for discrete time and time varing problems.**
- **They involve mostly solving linear equations and Matlab software is available.**
- **They are restricted to smooth systems with no state or control constraints.**
- **The LQR part must yield a Hurwitz $F + GK$.**
- **The software can used for systems of moderately large state dimension, e.g., $n = 25$, $m = 8$.**
- **Going to higher degree approximations to $\pi(x)$ and $\kappa(x)$ increases their accuracy near $x = 0$.**

Pros and Cons of Power Series Methods

- **Power Series Methods can also be used for discrete time and time varing problems.**
- **They involve mostly solving linear equations and Matlab software is available.**
- **They are restricted to smooth systems with no state or control constraints.**
- **The LQR part must yield a Hurwitz $F + GK$.**
- **The software can used for systems of moderately large state dimension, e.g., $n = 25$, $m = 8$.**
- **Going to higher degree approximations to $\pi(x)$ and $\kappa(x)$ increases their accuracy near $x = 0$.**
- **Going to higher degree approximations can enlarge the basin of stability of the closed loop system but it is not guaranteed. It can also decrease it.**

Pros and Cons of Power Series Methods

- **Power Series Methods can also be used for discrete time and time varing problems.**
- **They involve mostly solving linear equations and Matlab software is available.**
- **They are restricted to smooth systems with no state or control constraints.**
- **The LQR part must yield a Hurwitz $F + GK$.**
- **The software can used for systems of moderately large state dimension, e.g., $n = 25$, $m = 8$.**
- **Going to higher degree approximations to $\pi(x)$ and $\kappa(x)$ increases their accuracy near $x = 0$.**
- **Going to higher degree approximations can enlarge the basin of stability of the closed loop system but it is not guaranteed. It can also decrease it.**
- **Going to higher degree approximations requires more memory. There are $n + d - 1$ choose d monomials of degree d in n variables, approximately $n^d/d!$.**

Pros and Cons of Power Series Methods

- **Power Series Methods can also be used for discrete time and time varing problems.**
- **They involve mostly solving linear equations and Matlab software is available.**
- **They are restricted to smooth systems with no state or control constraints.**
- **The LQR part must yield a Hurwitz $F+GK$.**
- **The software can used for systems of moderately large state dimension, e.g., $n = 25$, $m = 8$.**
- **Going to higher degree approximations to $\pi(x)$ and $\kappa(x)$ increases their accuracy near $x = 0$.**
- **Going to higher degree approximations can enlarge the basin of stability of the closed loop system but it is not guaranteed. It can also decrease it.**
- **Going to higher degree approximations requires more memory. There are $n+d-1$ choose d monomials of degree d in n variables, approximately $n^d/d!$.**

Patchy Methods

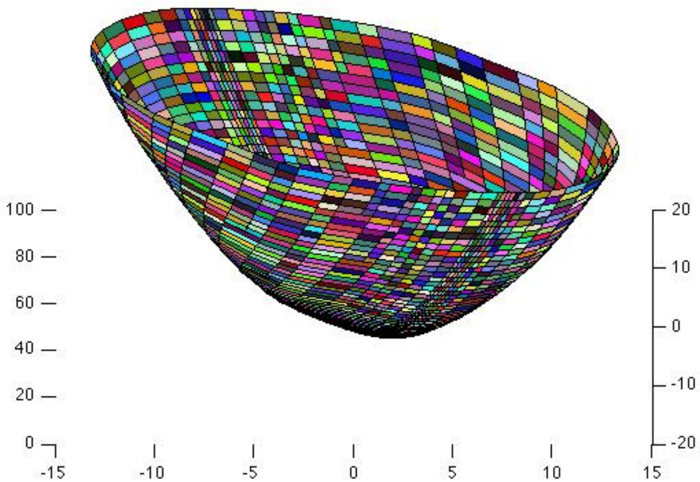


Figure : Optimal Cost of Inverting a Pendulum by a Torque at its Axis

Sequence of Patches

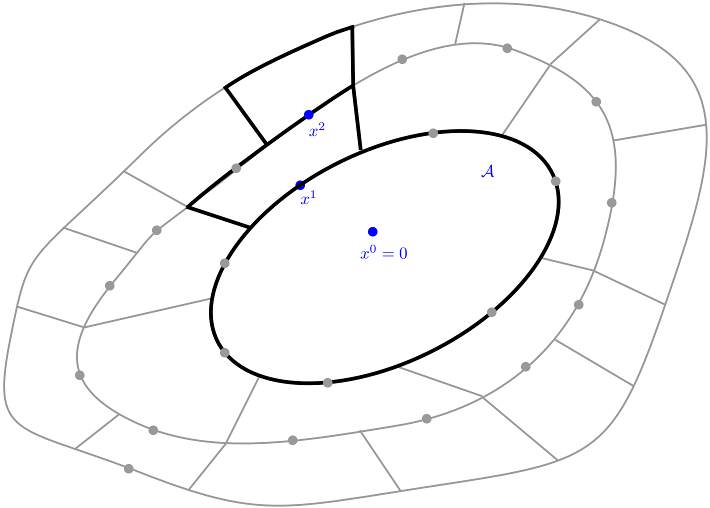


Figure : Sequence of Patches

Patch Calculation

The HJB equations are not singular away from the origin. The map

$$\pi^{[d+1]} \mapsto \frac{\partial \pi^{[d+1]}}{\partial x}(x) f(x^1, u^1)$$

takes a polynomial of degree $d+1$ to a polynomial of degree d.

Patch Calculation

The HJB equations are not singular away from the origin. The map

$$\pi^{[d+1]} \mapsto \frac{\partial \pi^{[d+1]}}{\partial x}(x) f(x^1, u^1)$$

takes a polynomial of degree $d+1$ to a polynomial of degree d.

So the map is not square. As a consequence $\pi^{[d+1]}(x)$ is not completely determined by the HJB equations.

Patch Calculation

The HJB equations are not singular away from the origin. The map

$$\pi^{[d+1]} \mapsto \frac{\partial \pi^{[d+1]}}{\partial x}(x) f(x^1, u^1)$$

takes a polynomial of degree $d+1$ to a polynomial of degree d.

So the map is not square. As a consequence $\pi^{[d+1]}(x)$ is not completely determined by the HJB equations.

Following Cauchy-Koveleskaya certain partial derivatives of $\pi(x)$ are inherited from partial derivatives of $\pi(x)$ on the previous patch.

Patch Calculation

The HJB equations are not singular away from the origin. The map

$$\pi^{[d+1]} \mapsto \frac{\partial \pi^{[d+1]}}{\partial x}(x) f(x^1, u^1)$$

takes a polynomial of degree $d+1$ to a polynomial of degree d.

So the map is not square. As a consequence $\pi^{[d+1]}(x)$ is not completely determined by the HJB equations.

Following Cauchy-Koveleskaya certain partial derivatives of $\pi(x)$ are inherited from partial derivatives of $\pi(x)$ on the previous patch.

For example we assume that $\frac{\partial \pi^1}{\partial x}(x^1) = z \frac{\partial \pi^0}{\partial x}(x^1)$ then the HJB equations reduce to a quadratic polynomial in the scalar z.

Patch Calculation

The HJB equations are not singular away from the origin. The map

$$\pi^{[d+1]} \mapsto \frac{\partial \pi^{[d+1]}}{\partial x}(x) f(x^1, u^1)$$

takes a polynomial of degree $d+1$ to a polynomial of degree d.

So the map is not square. As a consequence $\pi^{[d+1]}(x)$ is not completely determined by the HJB equations.

Following Cauchy-Koveleskaya certain partial derivatives of $\pi(x)$ are inherited from partial derivatives of $\pi(x)$ on the previous patch.

For example we assume that $\frac{\partial \pi^1}{\partial x}(x^1) = z \frac{\partial \pi^0}{\partial x}(x^1)$ then the HJB equations reduce to a quadratic polynomial in the scalar z.

Under suitable assumptions there is one positive root and one negative root. We take the positive root.

Invert a Pendulum

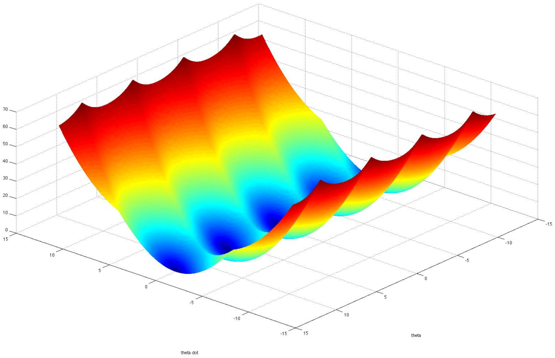


Figure : Periodicity of the Optimal Cost

The left axis is $-15 \leq \dot{\theta} \leq 15$ and the right axis is $-15 \leq \theta \leq 15$. From points on the ridges there are two optimal trajectories, one going to the left well and the other going to the right well.

Adaptive Algorithm

The algorithm is adaptive. It splits a patch in two when the relative residue of the first HJB equation is too high at the lower corners of a patch. It also lowers the upper level of a ring of patches if the relative residue is too high on it.

Ring	1	2	3	4
Initial Patch Level	0.64	1.21	1.96	2.89
Final Patch Level	0.36	0.63	1.38	2.23
Initial No. Patches	1	24	26	26
Final No. Patches	1	26	26	28

Adaptive Algorithm

The algorithm is adaptive. It splits a patch in two when the relative residue of the first HJB equation is too high at the lower corners of a patch. It also lowers the upper level of a ring of patches if the relative residue is too high on it.

Ring	1	2	3	4
Initial Patch Level	0.64	1.21	1.96	2.89
Final Patch Level	0.36	0.63	1.38	2.23
Initial No. Patches	1	24	26	26
Final No. Patches	1	26	26	28

The initial levels of the optimal cost were set at

$$(0.8)^2 \quad (1.1)^2 \quad (1.4)^2 \quad \ldots \quad (10.7)^2$$

Only the first ten patch levels were adjusted down.

Adaptive Algorithm

The algorithm is adaptive. It splits a patch in two when the relative residue of the first HJB equation is too high at the lower corners of a patch. It also lowers the upper level of a ring of patches if the relative residue is too high on it.

Ring	1	2	3	4
Initial Patch Level	0.64	1.21	1.96	2.89
Final Patch Level	0.36	0.63	1.38	2.23
Initial No. Patches	1	24	26	26
Final No. Patches	1	26	26	28

The initial levels of the optimal cost were set at

$$(0.8)^2 \quad (1.1)^2 \quad (1.4)^2 \quad \ldots \quad (10.7)^2$$

Only the first ten patch levels were adjusted down.

The last ring (34) contains 78 patches.

Patchy Hauser Osinga Pendulum

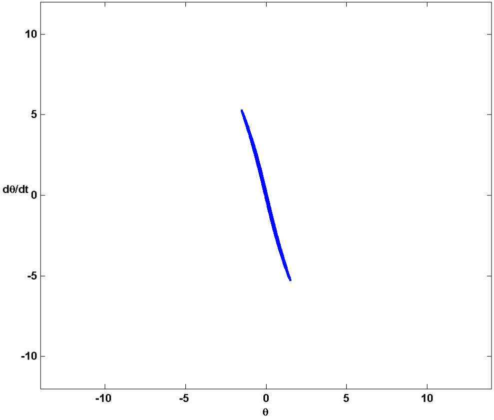


Figure : Patchy Optimal Cost to Level Set 70

Patchy Hauser Osinga Pendulum

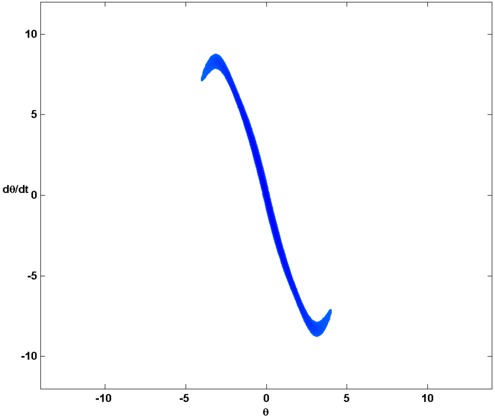


Figure : Patchy Optimal Cost to Level Set 140

Patchy Hauser Osinga Pendulum

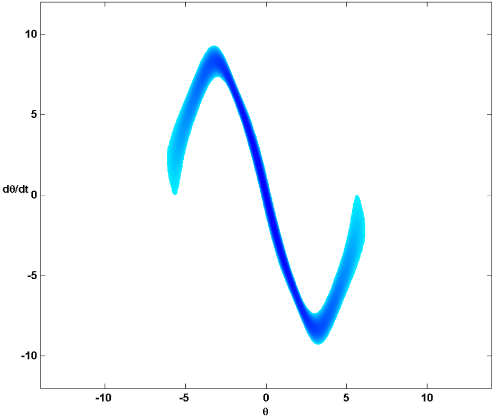


Figure : Patchy Optimal Cost to Level Set 210

Patchy Hauser Osinga Pendulum

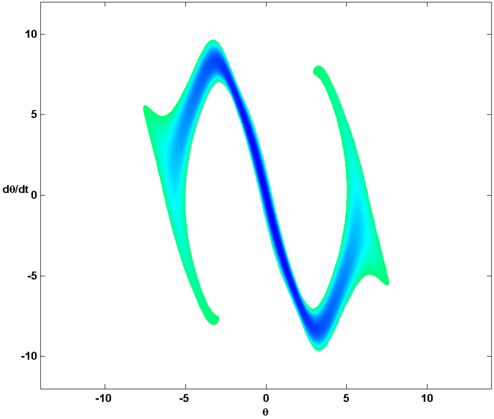


Figure : Patchy Optimal Cost to Level Set 280

Patchy Hauser Osinga Pendulum

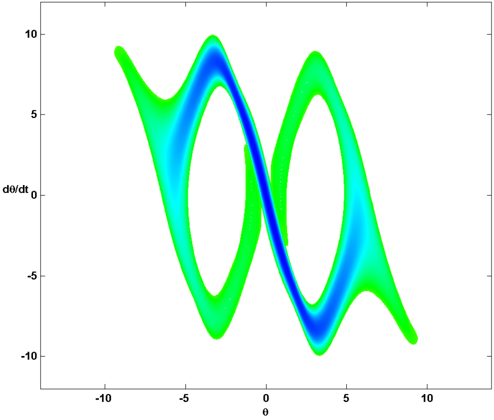


Figure : Patchy Optimal Cost to Level Set 350

Patchy Hauser Osinga Pendulum

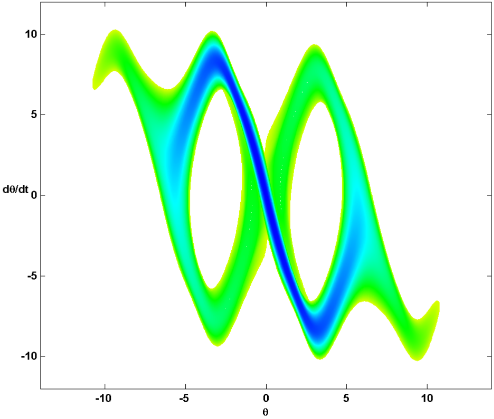


Figure : Patchy Optimal Cost to Level Set 420

Patchy Hauser Osinga Pendulum

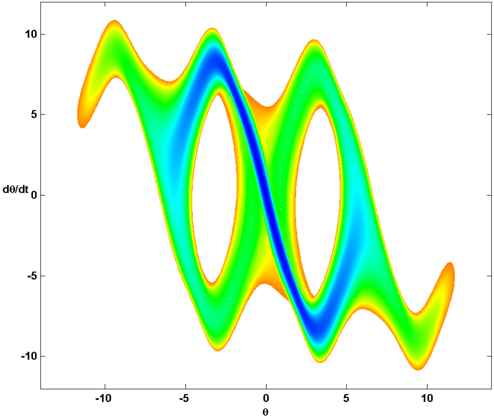


Figure : Patchy Optimal Cost to Level Set 490

Patchy Hauser Osinga Pendulum

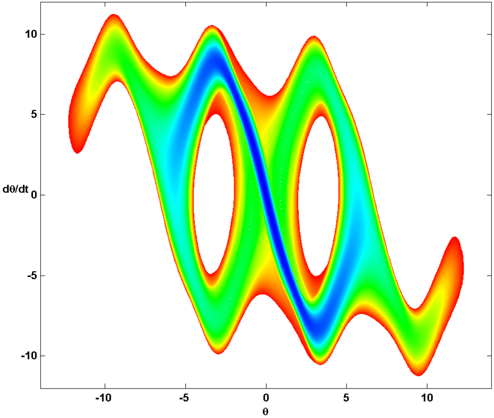


Figure : Patchy Optimal Cost to Level Set 566

Error Comparison

A nonlinear change of state coordinates on an LQR problem yields a nonlinear optimal control problem.

Error Comparison

A nonlinear change of state coordinates on an LQR problem yields a nonlinear optimal control problem.

The exact solution to the nonlinear problem is given by applying the nonlinear change of coordinates to the LQR solution.

Error Comparison

A nonlinear change of state coordinates on an LQR problem yields a nonlinear optimal control problem.

The exact solution to the nonlinear problem is given by applying the nonlinear change of coordinates to the LQR solution.

Here are the errors between the true optimal cost and the computed optimal cost which is of degree $d+1$.

	Max Error	Max Rel Error	Error Factor
LQR ($d=1$)	0.3543	0.8860	54.56
Al'brecht ($d=3$)	0.1636	0.4101	25.16
Patchy ($d=3$)	0.0065	0.0239	1

Error Comparison

A nonlinear change of state coordinates on an LQR problem yields a nonlinear optimal control problem.

The exact solution to the nonlinear problem is given by applying the nonlinear change of coordinates to the LQR solution.

Here are the errors between the true optimal cost and the computed optimal cost which is of degree $d+1$.

	Max Error	Max Rel Error	Error Factor
LQR $(d=1)$	$\mathbf{0.3543}$	$\mathbf{0.8860}$	$\mathbf{54.56}$
Al'brecht $(d=3)$	$\mathbf{0.1636}$	$\mathbf{0.4101}$	$\mathbf{25.16}$
Patchy $(d=3)$	$\mathbf{0.0065}$	$\mathbf{0.0239}$	$\mathbf{1}$

This shows that the patchy method can be very accurate and it is parallelizable.

Three Dimensional Example

Here is a level set of the patchy method applied to a three dimensional problem

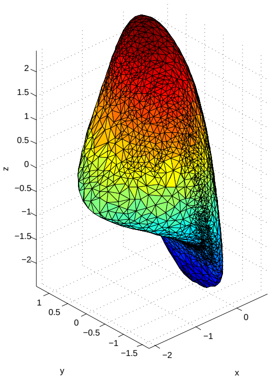


Figure : Level Set 55

Three Dimensional Example

Here is a level set of the patchy method applied to a three dimensional problem

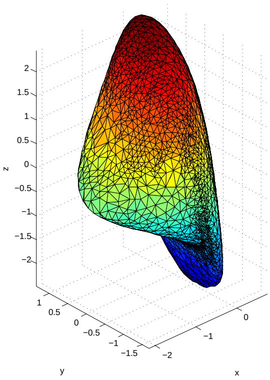


Figure : Level Set 55

The complexity of keeping track of the patches probably makes the patchy method infeasible in higher dimesions.

Conclusions

There are feasible methods for solving HJB or DP equations in dimensions $n = 2$ or $n = 3$.

Conclusions

There are feasible methods for solving HJB or DP equations in dimensions $n = 2$ or $n = 3$.

It is questionable whether any of these methods are feasible when $n = 4$ or $n = 5$.

Conclusions

There are feasible methods for solving HJB or DP equations in dimensions $n = 2$ or $n = 3$.

It is questionable whether any of these methods are feasible when $n = 4$ or $n = 5$.

It is unlikely that any of these methods are feasible when $n \geq 6$.

Conclusions

There are feasible methods for solving HJB or DP equations in dimensions $n = 2$ or $n = 3$.

It is questionable whether any of these methods are feasible when $n = 4$ or $n = 5$.

It is unlikely that any of these methods are feasible when $n \geq 6$.

Similar statements are true for the other PDEs of nonlinear control.

Trajectory Optimization

Typical Problem

$$\min_{u(0:T)} \int_0^T l(x,u)\ dt$$

subject to

$$\begin{aligned} \dot{x} &= f(x,u), &\quad 0 &\le g(x,u) \\ x(0) &= x^0, &\quad x(T) &= x^T \end{aligned}$$

Trajectory Optimization

Typical Problem

$$\min_{u(0:T)} \int_0^T l(x,u)\ dt$$

subject to

$$\dot{x} = f(x,u), \qquad 0 \le g(x,u)$$
$$x(0) = x^0, \qquad x(T) = x^T$$

Hamiltonian $\mathcal{H}(p,x,u) = pf(x,u) + l(x,u).$

Trajectory Optimization

Typical Problem

$$\min_{u(0:T)} \int_0^T l(x,u)\ dt$$

subject to

$$\dot{x} = f(x,u), \qquad 0 \le g(x,u)$$
$$x(0) = x^0, \qquad x(T) = x^T$$

Hamiltonian $\mathcal{H}(p,x,u) = pf(x,u) + l(x,u)$.

Pontryagin Minimum Principle: There exists $p(0:T) \in \mathbb{R}^{1\times n}$

Trajectory Optimization

Typical Problem

$$\min_{u(0:T)} \int_0^T l(x,u)\ dt$$

subject to

$$\begin{aligned} \dot{x} = f(x,u), &\qquad 0 \le g(x,u) \\ x(0) = x^0, &\qquad x(T) = x^T \end{aligned}$$

Hamiltonian $\mathcal{H}(p,x,u) = pf(x,u) + l(x,u)$.

Pontryagin Minimum Principle: There exists $p(0:T) \in I\!R^{1\times n}$

$$\begin{aligned} \dot{x}_i &= \frac{\partial \mathcal{H}}{\partial p_i}(p,x,u^*) \\ \dot{p}_i &= -\frac{\partial \mathcal{H}}{\partial x_i}(p,x,u^*) \\ u^* &= \operatorname{argmin}_u \{\mathcal{H}(p,x,u) : 0 \le g(x,u)\} \end{aligned}$$

plus boundary and transversality conditions.

Two Appoaches

Indirect Appoach: **Discretize the PMP equations and solve the resulting two point boundary value problem in $2n$ variables.**

Two Appoaches

Indirect Appoach: Discretize the PMP equations and solve the resulting two point boundary value problem in $2n$ variables.

This is usually done by shooting methods or multiple shooting methods.

Two Appoaches

Indirect Appoach: **Discretize the PMP equations and solve the resulting two point boundary value problem in $2n$ variables.**

This is usually done by shooting methods or multiple shooting methods. Such methods are slow and may not converge.

Two Appoaches

Indirect Appoach: **Discretize the PMP equations and solve the resulting two point boundary value problem in $2n$ variables.**

This is usually done by shooting methods or multiple shooting methods. Such methods are slow and may not converge.

Direct Appoach: **Discretize the trajectory optimization problem to get a nonlinear program which can be solved by existing software.**

Two Appoaches

Indirect Appoach: Discretize the PMP equations and solve the resulting two point boundary value problem in $2n$ variables.

This is usually done by shooting methods or multiple shooting methods. Such methods are slow and may not converge.

Direct Appoach: Discretize the trajectory optimization problem to get a nonlinear program which can be solved by existing software.

If we discretize time with step size h then decision variables are

$$u(0),\ u(h),\ u(2h), \ldots, u(T-h)$$

Regardless of the state dimension n it requires optimizing a function of mT/h variables subject to constraints.

Two Appoaches

Indirect Appoach: Discretize the PMP equations and solve the resulting two point boundary value problem in $2n$ variables.

This is usually done by shooting methods or multiple shooting methods. Such methods are slow and may not converge.

Direct Appoach: Discretize the trajectory optimization problem to get a nonlinear program which can be solved by existing software.

If we discretize time with step size h then decision variables are

$$u(0),\ u(h),\ u(2h), \ldots, u(T-h)$$

Regardless of the state dimension n it requires optimizing a function of mT/h variables subject to constraints.

Because of the development of excellent software for solving nonlinear programs, the direct approach has become more popular.

Discretization of the Optimal Trajectory Problem

$$
\begin{array}{rcl}
& \min\limits_{u(0:T)} \quad \sum_{t=0:h:T-1} & \bar{l}(x,u) \\
x^+ = \bar{f}(x,u), & & 0 \le g(x,u) \\
x(0) = x^0, & & x(T) = x^T
\end{array}
$$

where the discrete dynamics and discrete Lagrangian are defined by Lie differentiation

$$
\begin{aligned}
\bar{f}(x,u) &= x + f(x,u)h + L_{f(x,u)}f(x,u)\frac{h^2}{2} + L^2_{f(x,u)}f(x,u)\frac{h^3}{6} + \\
\bar{l}(x,u) &= l(x,u)h + L_{f(x,u)}l(x,u)\frac{h^2}{2} + L^2_{f(x,u)}l(x,u)\frac{h^3}{6} + \ldots
\end{aligned}
$$

Lie differentiation: $L_{f(x,u)}h(x,u) = \dfrac{\partial h}{\partial x}(x,u)f(x,u)$

Discretization of the Optimal Trajectory Problem

$$
\begin{array}{rcl}
 & \min_{u(0:T)} \quad \sum_{t=0:h:T-1} & \bar{l}(x,u) \\
x^+ = \bar{f}(x,u), & & 0 \le g(x,u) \\
x(0) = x^0, & & x(T) = x^T
\end{array}
$$

where the discrete dynamics and discrete Lagrangian are defined by Lie differentiation

$$
\begin{aligned}
\bar{f}(x,u) &= x + f(x,u)h + L_{f(x,u)} f(x,u)\frac{h^2}{2} + L^2_{f(x,u)} f(x,u)\frac{h^3}{6} + \\
\bar{l}(x,u) &= l(x,u)h + L_{f(x,u)} l(x,u)\frac{h^2}{2} + L^2_{f(x,u)} l(x,u)\frac{h^3}{6} + \ldots
\end{aligned}
$$

Lie differentiation: $L_{f(x,u)} h(x,u) = \dfrac{\partial h}{\partial x}(x,u) f(x,u)$

The Euler approximation stops at the h terms.

Discretization of the Optimal Trajectory Problem

$$
\begin{array}{rcl}
 & \min_{u(0:T)} \quad \sum_{t=0:h:T-1} & \bar{l}(x,u) \\
x^+ = \bar{f}(x,u), & & 0 \le g(x,u) \\
x(0) = x^0, & & x(T) = x^T
\end{array}
$$

where the discrete dynamics and discrete Lagrangian are defined by Lie differentiation

$$
\begin{aligned}
\bar{f}(x,u) &= x + f(x,u)h + L_{f(x,u)} f(x,u)\frac{h^2}{2} + L^2_{f(x,u)} f(x,u)\frac{h^3}{6} + \\
\bar{l}(x,u) &= l(x,u)h + L_{f(x,u)} l(x,u)\frac{h^2}{2} + L^2_{f(x,u)} l(x,u)\frac{h^3}{6} + \dots
\end{aligned}
$$

Lie differentiation: $L_{f(x,u)}h(x,u) = \frac{\partial h}{\partial x}(x,u)f(x,u)$

The Euler approximation stops at the h terms.

If Lie differentiation is difficult use Runge-Kutta approximations.

Discretization of the Optimal Trajectory Problem

There are other discretization schemes. The goal of any such scheme is to approximate the continuous dynamics with high accuracy using as few node points as possible.

Discretization of the Optimal Trajectory Problem

There are other discretization schemes. The goal of any such scheme is to approximate the continuous dynamics with high accuracy using as few node points as possible.

In the scheme described on the last side accuracy is achieved by using a higher order method and a small stepsize h . The number of nodes is T/h so small h leads to a large number of nodes.

Discretization of the Optimal Trajectory Problem

There are other discretization schemes. The goal of any such scheme is to approximate the continuous dynamics with high accuracy using as few node points as possible.

In the scheme described on the last side accuracy is achieved by using a higher order method and a small stepsize h . The number of nodes is T/h so small h leads to a large number of nodes.

Recall that we have to find the minimum of a function of mT/h variables, $u(0), u(h), u(2h), \ldots, u(T-1)$.

Discretization of the Optimal Trajectory Problem

There are other discretization schemes. The goal of any such scheme is to approximate the continuous dynamics with high accuracy using as few node points as possible.

In the scheme described on the last side accuracy is achieved by using a higher order method and a small stepsize h . The number of nodes is T/h so small h leads to a large number of nodes.

Recall that we have to find the minimum of a function of mT/h variables, $u(0), u(h), u(2h), \ldots, u(T-1)$.

The discretization of the continuous time problem is a form a quadrature so we could use any quadrature rule, e.g., Euler, Trapezoidal, Hermite-Simpson, etc. in either explicit or implicit form.

Efficient Quadrature Rules

Perhaps the most efficient quadrature is Legendre-Gauss (LG). It uses only N nodes to exactly integrate any polynomial of degree $2N-1$ or less.

Efficient Quadrature Rules

Perhaps the most efficient quadrature is Legendre-Gauss (LG). It uses only N nodes to exactly integrate any polynomial of degree $2N - 1$ or less.

On the standard interval $[-1, 1]$ it takes the form

$$\int_{-1}^{1} \phi(t)\ dt \ = \ \sum_{i=1}^{N} w_i \phi(t_i)$$

where the nodes t_i are the zeros of the N^{th} Legendre polynomial $P_N(t)$.

Efficient Quadrature Rules

Perhaps the most efficient quadrature is Legendre-Gauss (LG). It uses only N nodes to exactly integrate any polynomial of degree $2N-1$ or less.

On the standard interval $[-1,1]$ it takes the form

$$\int_{-1}^{1} \phi(t)\ dt \;=\; \sum_{i=1}^{N} w_i \phi(t_i)$$

where the nodes t_i are the zeros of the N^{th} Legendre polynomial $P_N(t)$.

The weights are given by

$$w_i \;=\; \frac{2}{(1-t_i^2)(P_N'(t_i))^2}$$

Efficient Quadrature Rules

Perhaps the most efficient quadrature is Legendre-Gauss (LG). It uses only N nodes to exactly integrate any polynomial of degree $2N-1$ or less.

On the standard interval $[-1,1]$ it takes the form

$$\int_{-1}^{1} \phi(t)\, dt \;=\; \sum_{i=1}^{N} w_i \phi(t_i)$$

where the nodes t_i are the zeros of the N^{th} Legendre polynomial $P_N(t)$.

The weights are given by

$$w_i \;=\; \frac{2}{(1-t_i^2)(P_N'(t_i))^2}$$

But all the nodes t_i are in the open interval $(-1,1)$ so Legendre-Gauss quadrature is not suitable if there are boundary conditions.

Efficient Quadrature Rules

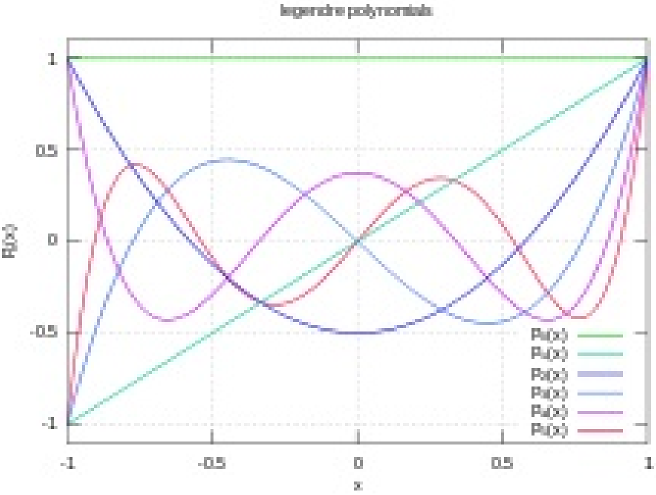


Figure : Legendre Polynomials to Degree 5

Efficient Quadrature Rules

Legendre-Gauss-Lobatto (LGL) quadrature is slightly less efficient, It uses $N+1$ nodes to exactly integrate any polynomial of degree $2N-1$ or less.

Efficient Quadrature Rules

Legendre-Gauss-Lobatto (LGL) quadrature is slightly less efficient, It uses $N+1$ nodes to exactly integrate any polynomial of degree $2N-1$ or less.

It uses the two endpoints $t_0 = -1,\ t_N = 1$ and the $N-1$ zeros $t_2, \ldots, t_{N-1}$ of $P'_N(t)$.

Efficient Quadrature Rules

Legendre-Gauss-Lobatto (LGL) quadrature is slightly less efficient, It uses $N+1$ nodes to exactly integrate any polynomial of degree $2N-1$ or less.

It uses the two endpoints $t_0 = -1,\ t_N = 1$ and the $N-1$ zeros $t_2, \ldots, t_{N-1}$ of $P_N'(t)$.

The weights are $\frac{2}{n(n-1)}$ at the endpoints and

$$w_i = \frac{2}{n(n-1)(P_N(t_i))^2}$$

in between.

Pseudospectral Trajectory Optimization

Gong, Kang and Ross have shown that the pseudospectral method converges for feedback linearizable systems. If $m = 1$ such a system can be transformed to

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 \\
&\vdots \\
\dot{x}_{n-1} &= x_n \\
\dot{x}_n &= f_n(x) + g_n(x)u
\end{aligned}$$

Pseudospectral Trajectory Optimization

Gong, Kang and Ross have shown that the pseudospectral method converges for feedback linearizable systems. If $m = 1$ such a system can be transformed to

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 \\
&\vdots \\
\dot{x}_{n-1} &= x_n \\
\dot{x}_n &= f_n(x) + g_n(x)u
\end{aligned}$$

Minimize

$$\int_{-1}^{1} l(x(t), u(t))\, dt + \alpha(x(-1), x(1))$$

subject to

$$\begin{aligned}
0 &= \beta(x(-1), x(1)) \\
0 &\leq \gamma(x(t), u(t))
\end{aligned}$$

Pseudospectral Trajectory Optimization

Each $x_i(t)$ is approximated by an N^{th} degree interpolating polynomial $\bar{x}_i(t)$. These polynomials are represented by their values at the $N+1$ LGL nodes,

$$\begin{aligned} \bar{x}_i &= \left[\begin{array}{ccc} \bar{x}_i^0 & \dots & \bar{x}_i^N \end{array} \right]' \\ \bar{x}_i(t) &= \sum_0^N \bar{x}_i^j \phi_j(t) \end{aligned}$$

where the $\phi_j(t)$ are the Lagrange polynomials at the LGL nodes.

Pseudospectral Trajectory Optimization

Each $x_i(t)$ is approximated by an N^{th} degree interpolating polynomial $\bar{x}_i(t)$. These polynomials are represented by their values at the $N+1$ LGL nodes,

$$\begin{aligned} \bar{x}_i &= \begin{bmatrix} \bar{x}_i^0 & \ldots & \bar{x}_i^N \end{bmatrix}' \\ \bar{x}_i(t) &= \sum_0^N \bar{x}_i^j \phi_j(t) \end{aligned}$$

where the $\phi_j(t)$ are the Lagrange polynomials at the LGL nodes.

The dynamics is approximated by the equations

$$\begin{aligned} \bar{x}_{i+1} &= D\bar{x}_i, \qquad i = 1, \ldots, n-1 \\ \bar{u}^j &= \frac{(D\bar{x}_n)^j - f(\bar{x}^j)}{g(\bar{x}^j)} \end{aligned}$$

so the $N+1$ decision variables are $\bar{x}_1^0, \ldots, \bar{x}_1^N$.

Multiplication of the interpolated values of a polynomial by the differentiation matrix D yields the interpolating values of its derivative.

Multiplication of the interpolated values of a polynomial by the differentiation matrix D yields the interpolating values of its derivative.

The control $u(t)$ is not an interpolating polynomial, rather

$$u(t) = \frac{\dot{\bar{x}}_n - f(\bar{x}(t))}{g(\bar{x}(t))}$$

Pseudospectral Trajectory Optimization

Multiplication of the interpolated values of a polynomial by the differentiation matrix D yields the interpolating values of its derivative.

The control $u(t)$ is not an interpolating polynomial, rather

$$u(t) = \frac{\dot{\bar{x}}_n - f(\bar{x}(t))}{g(\bar{x}(t))}$$

The cost is approximated by a LGL quadrature

$$\sum_{j=0}^{N} l(x^j, u^j) w_j + \alpha(\bar{x}^0, \bar{x}^N)$$

The boundary conditions are approximated by a relaxed version of

$$0 = \beta(\bar{x}^0, \bar{x}^N)$$

and the constraints are approximated by a relaxed version of

$$0 \le \gamma(\bar{x}^j, \bar{u}^j), \qquad j = 0, \dots, N$$

Pseudospectral Trajectory Optimization and the ISS

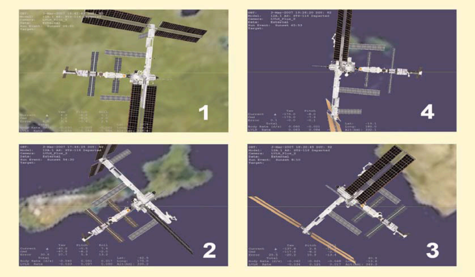
SIAM News, Volume 40, Number 7, September 2007

Pseudospectral Trajectory Optimization and the ISS

SIAM News, Volume 40, Number 7, September 2007

Pseudospectral Optimal Control Theory Makes Debut Flight, Saves NASA $1M in Under Three Hours

By Wei Kang and Naz Bedrossian



Model Predictive Control

Suppose the problem of minimizing

$$\int_0^\infty l(x,u)\ dt$$

subject to

$$\begin{aligned} \dot{x} &= f(x,u) \\ x(0) &= x^0 \\ 0 &\le g(x,u) \end{aligned}$$

has been discretized into minimizing

$$\sum_{t=0:h:\infty} \bar{l}(x(t),u(t))$$

subject to

$$\begin{aligned} x^+ &= \bar{f}(x,u) \\ x(0) &= x^0 \\ 0 &\le g(x,u) \end{aligned}$$

Model Predictive Control

Minimization over the infinite horizon is too difficult so choose a time window T and a terminal cost $\pi_T(x)$ defined on a terminal set $\mathcal{X}_T$ which is a compact neighborhood of $x = 0$.

Model Predictive Control

Minimization over the infinite horizon is too difficult so choose a time window T and a terminal cost $\pi_T(x)$ defined on a terminal set $\mathcal{X}_T$ which is a compact neighborhood of $x = 0$.

Consider the problem of minimizing

$$\sum_{t=0:h:T-h} \bar{l}(x(t), u(t)) + \pi_T(x(T))$$

subject to

$$\begin{aligned} x^+ &= \bar{f}(x, u) \\ x(0) &= x^0 \\ 0 &\le g(x, u) \\ x(T) &\in \mathcal{X}_T \end{aligned}$$

Model Predictive Control

Minimization over the infinite horizon is too difficult so choose a time window T and a terminal cost $\pi_T(x)$ defined on a terminal set $\mathcal{X}_T$ which is a compact neighborhood of $x = 0$.

Consider the problem of minimizing

$$\sum_{t=0:h:T-h} \bar{l}(x(t), u(t)) + \pi_T(x(T))$$

subject to

$$\begin{array}{rcl} x^+ & = & \bar{f}(x,u) \\ x(0) & = & x^0 \\ 0 & \leq & g(x,u) \\ x(T) & \in & \mathcal{X}_T \end{array}$$

The decision variables are $u(0), \ldots, u(T-h)$.

Model Predictive Control

Then pass this nonlinear program to a fast solver to find the optimal $u^0(0), \dots, u^0(T-h)$. This needs to be done in less than the time step h.

Model Predictive Control

Then pass this nonlinear program to a fast solver to find the optimal $u^0(0), \ldots, u^0(T-h)$. This needs to be done in less than the time step h.

Use the control $u^0(0)$ to get the state to $x^1 = x(h)$.

Model Predictive Control

Then pass this nonlinear program to a fast solver to find the optimal $u^0(0), \dots, u^0(T-h)$. This needs to be done in less than the time step h.

Use the control $u^0(0)$ to get the state to $x^1 = x(h)$.

Then between times h and $2h$ solve the problem of minimizing

$$\sum_{t=h:h:T} \bar{l}(x(t), u(t)) + \pi_T(x(T+h))$$

subject to

$$\begin{array}{rcl} x^+ & = & \bar{f}(x,u) \\ x(h) & = & x^1 \\ 0 & \leq & g(x,u) \\ x(T+h) & \in & \mathcal{X}_T \end{array}$$

to obtain the optimal $u^1(h), \dots, u^1(T)$.

Model Predictive Control

Then pass this nonlinear program to a fast solver to find the optimal $u^0(0), \ldots, u^0(T-h)$. This needs to be done in less than the time step h.

Use the control $u^0(0)$ to get the state to $x^1 = x(h)$.

Then between times h and $2h$ solve the problem of minimizing

$$\sum_{t=h:h:T} \bar{l}(x(t), u(t)) + \pi_T(x(T+h))$$

subject to

$$\begin{array}{rcl} x^+ & = & \bar{f}(x,u) \\ x(h) & = & x^1 \\ 0 & \leq & g(x,u) \\ x(T+h) & \in & \mathcal{X}_T \end{array}$$

to obtain the optimal $u^1(h), \ldots, u^1(T)$.

Use the control $u^1(h)$ to get the state to $x^2 = x(2h)$, etc.

Model Predictive Control

The key issues are the following

- **If the discrete time system is a discretization of a continuous time system then the time step h must be short enough to accurately approximate it.**

Model Predictive Control

The key issues are the following

- **If the discrete time system is a discretization of a continuous time system then the time step h must be short enough to accurately approximate it.**
- **The time step h should be long enough for the nonlinear program to be solved in one time step.**

Model Predictive Control

The key issues are the following

- **If the discrete time system is a discretization of a continuous time system then the time step h must be short enough to accurately approximate it.**
- **The time step h should be long enough for the nonlinear program to be solved in one time step.**
- **The horizon T must be short enough so that the nonlinear program can be solved in one time step h.**

Model Predictive Control

The key issues are the following

- **If the discrete time system is a discretization of a continuous time system then the time step h must be short enough to accurately approximate it.**
- **The time step h should be long enough for the nonlinear program to be solved in one time step.**
- **The horizon T must be short enough so that the nonlinear program can be solved in one time step h.**
- **The horizon T must be long enough and/or $\mathcal{X}_T$ large enough so that $x(t+T) \in \mathcal{X}_T$.**

Model Predictive Control

The key issues are the following

- **If the discrete time system is a discretization of a continuous time system then the time step h must be short enough to accurately approximate it.**
- **The time step h should be long enough for the nonlinear program to be solved in one time step.**
- **The horizon T must be short enough so that the nonlinear program can be solved in one time step h.**
- **The horizon T must be long enough and/or $\mathcal{X}_T$ large enough so that $x(t+T) \in \mathcal{X}_T$.**
- **The initial guess of $u^0(0), \ldots, u^0(T-1)$ that is fed to the solver must be close to optimal else the solver may fail to converge to the true solution.**

Model Predictive Control

- **This is not as much a problem with later initial guesses because we can take $u^0(h), \ldots, u^0(T-h)$ as the initial guess for $u^1(h), \ldots, u^1(T-h)$.**

Model Predictive Control

- **This is not as much a problem with later initial guesses because we can take $u^0(h), \ldots, u^0(T-h)$ as the initial guess for $u^1(h), \ldots, u^1(T-h)$.**
- **The ideal terminal cost $\pi_T(x)$ is the optimal cost of the infinite horizon optimal control problem provided that it can be computed on a large enough $\mathcal{X}_T$. Then the exact solutions to the finite horizon and infinite horizon optimal control problems are identical.**

Model Predictive Control

- **This is not as much a problem with later initial guesses because we can take $u^0(h), \ldots, u^0(T-h)$ as the initial guess for $u^1(h), \ldots, u^1(T-h)$.**
- **The ideal terminal cost $\pi_T(x)$ is the optimal cost of the infinite horizon optimal control problem provided that it can be computed on a large enough $\mathcal{X}_T$. Then the exact solutions to the finite horizon and infinite horizon optimal control problems are identical.**
- **If the infinite horizon optimal control law $\kappa_T(x)$ is known on the terminal set $\mathcal{X}_T$ then the initial guess for $u^1(T)$ should be $\kappa_T(x^0(T))$ where $\bar{x}^0(T)$ is the T^{th} state generated by the last control sequence. $u^0(0), \ldots, u^0(T-1)$**

Concluding Remarks

- **We can solve HJB or Dynamic Programming Equations only in low state dimensions.**

Concluding Remarks

- **We can solve HJB or Dynamic Programming Equations only in low state dimensions.**
- **Pseudospectral methods can solve trajectory optimization problems in low to medium state dimensions.**

Concluding Remarks

- **We can solve HJB or Dynamic Programming Equations only in low state dimensions.**
- **Pseudospectral methods can solve trajectory optimization problems in low to medium state dimensions.**
- **Model Predictive Control is a viable alternative to solving Dynamic Programming Equations in low to medium state dimensions for slow processes even when there are state and control constraints.**

Concluding Remarks

- **We can solve HJB or Dynamic Programming Equations only in low state dimensions.**
- **Pseudospectral methods can solve trajectory optimization problems in low to medium state dimensions.**
- **Model Predictive Control is a viable alternative to solving Dynamic Programming Equations in low to medium state dimensions for slow processes even when there are state and control constraints.**
- **It may be possible to use power series methods to compute the terminal cost $\pi_T(x)$ and feedback $\kappa_T(x)$ on a larger terminal set $\mathcal{X}_T$. This may allow us to lengthen the time step h and/or shorten the horizon T so that MPC can be used on faster processes.**

Concluding Remarks

- **We can solve HJB or Dynamic Programming Equations only in low state dimensions.**
- **Pseudospectral methods can solve trajectory optimization problems in low to medium state dimensions.**
- **Model Predictive Control is a viable alternative to solving Dynamic Programming Equations in low to medium state dimensions for slow processes even when there are state and control constraints.**
- **It may be possible to use power series methods to compute the terminal cost $\pi_T(x)$ and feedback $\kappa_T(x)$ on a larger terminal set $\mathcal{X}_T$. This may allow us to lengthen the time step h and/or shorten the horizon T so that MPC can be used on faster processes.**
- **For a copy of these slides contact** `ajkrener@nps.edu`

Concluding Remarks

- **We can solve HJB or Dynamic Programming Equations only in low state dimensions.**
- **Pseudospectral methods can solve trajectory optimization problems in low to medium state dimensions.**
- **Model Predictive Control is a viable alternative to solving Dynamic Programming Equations in low to medium state dimensions for slow processes even when there are state and control constraints.**
- **It may be possible to use power series methods to compute the terminal cost $\pi_T(x)$ and feedback $\kappa_T(x)$ on a larger terminal set $\mathcal{X}_T$. This may allow us to lengthen the time step h and/or shorten the horizon T so that MPC can be used on faster processes.**
- **For a copy of these slides contact** ajkrener@nps.edu
- **Thank you! Questions?**

Bibliography

- M. Bardi, I. Caputo-Dolcetta, Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations, Birkhauser, 1997.
- M. Falcone, Numerical Solution of Dynamic Programming Equations, Appendix A, Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations, Birkhauser, 1997.
- M. Boue and P. Dupuis, Markov Chain Approximations for Deterministic Control Problems with Affine Dynamics and Quadratic Cost in the Control, SIAM J. Numerical Analysis, v. 36, pp. 667-695, 1999.
- J. N. Tsitsiklis, Efficient Algorithms for Globally Optimal Trajectories, IEEE Transactions on Automatic Control, v. 40, pp. 1528-1538, 1995.

Bibliography

- J. A. Sethian, A fast marching level set method for monotonically advancing fronts, Proc.Natl. Acad. Sci. USA, 93 (1996), 1591-1595.
- J. A. Sethian and A. Vladimirsky, Ordered upwind methods for static Hamilton-Jacobi equations: theory and algorithms, SIAM J. Numer. Anal., 41 (2003), 325-363.
- E. Carlini, M. Falcone, N. Forcadel, and R. Monneau, Conver- gence of a Generalized Fast Marching Method for an Eikonal equation with a velocity changing sign, SIAM J. Numer. Anal. v. 46 , pp. 2920-2952, 2008.
- J. Hauser and H. Osinga On the Geometry of Optimal Control: the Inverted Pendulum Example, in Proceedings of the American Control Conference, Arlington VA, USA, Volume 2, pp. 17211726, 2001.

Bibliography

- M. Akian, S. Gaubert and A. Lakhoua, A max-plus finite element method for solving finite horizon determinsitic optimal control problems, Proc. 16th International Symposium on Mathematical Theory of Networks and Systems (MTNS), Leuven, 2004.
- W. McEneaney, A Curse of Dimensionality Free Method for the Solution of Certain HJB PDEs, SIAM J. Control and Opt. (2007).
- A. Szpiro and P. Dupuis, A Second Order Numerical Method for First Order Hamilton-Jacobi Equations, SIAM J. Numer. Anal., v. 40, pp.1136-1183, 2002.

Bibliography

- E. G. Al'brekht, On the Optimal Stabilization of Nonlinear Systems, J. Appl. Math. Mech., v. 25, pp. 1254-1266, 1961.
- C. Navasca and A. J. Krener, Patchy Solutions of Hamilton-Jacobi-Bellman Partial Differential Equations, in A. Chiuso et al. (eds.), Modeling, Estimation and Control, Lecture Notesin Control and Information Sciences, 364 (2007), 251-270.
- T. Hunt, A proof of the higher order accuracy of the patchy method for solving the Hamilton-Jacobi-Bellman equation, Ph.D. Thesis, University of California, Davis, 2011.
- S. Cacace, E. Cristiani, M. Falcone and A. Picarelli, A patchy dynamic pro- gramming scheme for a class of Hamilton-Jacobi-Bellman equations, accepted by SIAM Journal on Scientic Computing.

Bibliography

- C. O. Aguilar, C. O., T. W. Hunt and A. J. Krener, An Adaptive Patchy Method for Solving Hamilton Jacobi Bellman Equations, Proceedings of the International Conference of Numerical Analysis and Applied Mathematics, American Institute of Physics Conference Proceedings no. 1479, 2012.
- C. O. Aguilar and A. J. Krener, Patchy Solution of a Francis-Byrnes-Isidori Partial Diffential Equation, International Journal of Robust and Nonlinear Control,
- B. Krauskopf, H.M. Osinga, E.J. Doedel, M.E. Henderson, J. Guckenheimer, A. Vladimirsky, M. Dellnitz and O. Junge, A survey of methods for computing (un)stable manifolds of vector fields, Int. J. Bifurcation and Chaos, v. 15, pp. 763-791, 2005.

Bibliography

- **M. Gerdts, Numerical Optimal Control, OMPC 2013 - SADCO Summer School and Workshop on Optimal and Model Predictive Control. Slides available at http://num.math.uni-bayreuth.de/en/conferences/**

- **F. Fahroo and I.M. Ross. Direct Trajectory Optimization by a Chebyshev Pseudospectral Method. Journal of Guidance Control and Dynamics, 25, 2002.**

- **Q. Gong, W. Kang and I. M. Ross, A Pseudospectral Method for the Optimal Control of Constrained Feedback Linearizable Systems, IEEE Trans. on Automatic Control, v. 51, pp.1115-1129, 2006**

- **D. Q. Mayne, J. B. Rawlings, C. V. Rao and P. O. M. Scokaert Constrained model predictive control: Stability and optimality. Automatica, v. 36, pp.789-814, 2000.**

Bibliography

- J. B. Rawlings and D. Q. Mayne. Model Predictive Control: Theory and Design. Nob Hill Publishing, Madison, WI, 2009.
- J. B. Rawlings, Model Predictve Control, OMPC 2013 - SADCO Summer School and Workshop on Optimal and Model Predictive Control. Slides available at http://num.math.uni-bayreuth.de/en/conferences/