Speeding up Model Predictive Control via Al'brekht's Method and its Extensions

Arthur J. Krener

ajkrener@ucdavis.edu

Research supported by AFOSR and NSF

Al' brekht's Method

Here is the key to speeding up Model Predictive Control

E. G. Al'brekht, On the Optimal Stabilization of Nonlinear Systems, J. Appl. Math. Mech., v. 25, pp. 1254-1266, 1961.

This paper is Albrekht's first publication.

Al' brekht's Method

Here is the key to speeding up Model Predictive Control

E. G. Al'brekht, On the Optimal Stabilization of Nonlinear Systems, J. Appl. Math. Mech., v. 25, pp. 1254-1266, 1961.

This paper is Albrekht's first publication.

He was first a student and then later a close asssociate of N. N. Krasovski's so I assume that Krasovski had something to do with it.

Al' brekht's Method

Here is the key to speeding up Model Predictive Control

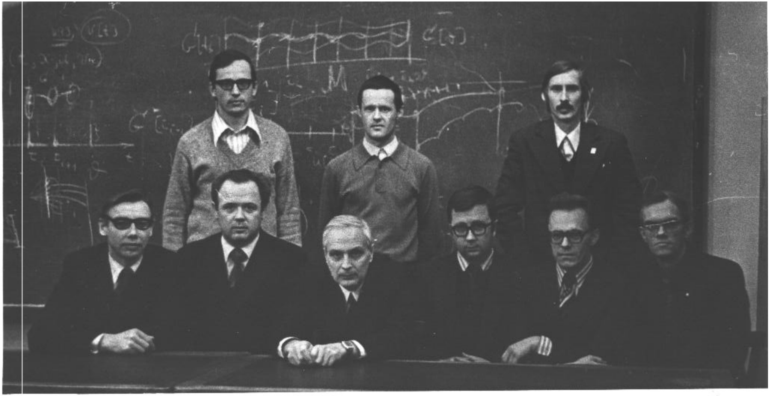
E. G. Al'brekht, On the Optimal Stabilization of Nonlinear Systems, J. Appl. Math. Mech., v. 25, pp. 1254-1266, 1961.

This paper is Albrekht's first publication.

He was first a student and then later a close asssociate of N. N. Krasovski's so I assume that Krasovski had something to do with it.

Альбрехт Э.Г. Об оптимальной стабилизации нелинейных систем. ПММ, т. 25, вып. 5, 1961, с. 836 - 844.

E. G. Al'brekht and N. N. Krasovski



Optimal Stabilization

Notice that Al'brekht's goal is Stabilization.

Optimal Stabilization

Notice that Al'brekht's goal is Stabilization.

Optimization is a means to this goal.

Optimal Stabilization

Notice that Al'brekht's goal is Stabilization.

Optimization is a means to this goal.

This shows that Al'brekht was working in the long tradition of studying stability and stabilizability following A. M. Lyapunov, N. G. Chetayev and N. N. Krasovski.

Optimal Stabilization

Notice that Al'brekht's goal is Stabilization.

Optimization is a means to this goal.

This shows that Al'brekht was working in the long tradition of studying stability and stabilizability following A. M. Lyapunov, N. G. Chetayev and N. N. Krasovski.

He was also influenced by the optimization techniques of L. S. Pontryagin and R. E. Bellman.

Stabilization Problem

Given a nonlinear system

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

Stabilization Problem

Given a nonlinear system

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

and an equilibrium point

$$0 = f(x^e, u^e)$$

Stabilization Problem

Given a nonlinear system

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

and an equilbrium point

$$0 = f(x^e, u^e)$$

find a feedback

$$u = \kappa(x)$$

Stabilization Problem

Given a nonlinear system

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

and an equilbrium point

$$0 = f(x^e, u^e)$$

find a feedback

$$u = \kappa(x)$$

so that the closed loop system

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

Stabilization Problem

Given a nonlinear system

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

and an equilbrium point

$$0 = f(x^e, u^e)$$

find a feedback

$$u = \kappa(x)$$

so that the closed loop system

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

is (at least locally) asymptotically stable around x^e.

Stabilization Problem

Given a nonlinear system

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

and an equilbrium point

$$0 = f(x^e, u^e)$$

find a feedback

$$u = \kappa(x)$$

so that the closed loop system

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

is (at least locally) asymptotically stable around x^e.

WLOG $x^e = 0, \ u^e = 0$ **.**

Infinite Horizon Optimal Control Problem

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

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

$$x \in I\!R^{n\times 1}, \qquad u \in I\!R^{m\times 1}$$

Infinite Horizon Optimal Control Problem

$$\min_{u(0:\infty)} \int_0^\infty l(x,u)\ dt$$
$$\dot{x} = f(x,u), \qquad x(0) = x^0$$
$$x \in \mathbb{R}^{n\times 1}, \qquad u \in \mathbb{R}^{m\times 1}$$

Optimal Cost $\pi(x)$ and Optimal Feedback $u = \kappa(x)$

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

Infinite Horizon Optimal Control Problem

$$\min_{u(0:\infty)} \int_0^\infty l(x,u)\ dt$$
$$\dot{x} = f(x,u), \qquad x(0) = x^0$$
$$x \in I\!R^{n\times 1}, \qquad u \in I\!R^{m\times 1}$$

Optimal Cost $\pi(x)$ and Optimal Feedback $u = \kappa(x)$

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

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

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

Infinite Horizon Optimal Control Problem

$$\min_{u(0:\infty)} \int_0^\infty l(x,u)\;dt$$
$$\dot{x} = f(x,u), \qquad x(0) = x^0$$
$$x \in I\!R^{n\times 1}, \qquad u \in I\!R^{m\times 1}$$

Optimal Cost $\pi(x)$ and Optimal Feedback $u = \kappa(x)$

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

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

$$0 \;=\; \mathcal{H}\left(\frac{\partial \pi}{\partial x}(x), x, \kappa(x)\right)$$
$$\kappa(x) \;=\; \text{argmin}_u \mathcal{H}\left(\frac{\partial \pi}{\partial x}(x), x, u\right)$$

Lyapunov Argument

If the optimal cost $\pi(x)$ and optimal feedback $\kappa(x)$ can be found then a basin of attraction can be verified by a Lyapunov argument.

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

Lyapunov Argument

If the optimal cost $\pi(x)$ and optimal feedback $\kappa(x)$ can be found then a basin of attraction can be verified by a Lyapunov argument.

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

If

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

then stability can be verified by a Krasovski-Barbashin argument.

Lyapunov Argument

If the optimal cost $\pi(x)$ and optimal feedback $\kappa(x)$ can be found then a basin of attraction can be verified by a Lyapunov argument.

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

If

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

then stability can be verified by a Krasovski-Barbashin argument.

So optimal control can be used for finding and verifying stabilizing feedbacks.

Lyapunov Argument

If the optimal cost $\pi(x)$ and optimal feedback $\kappa(x)$ can be found then a basin of attraction can be verified by a Lyapunov argument.

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

If

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

then stability can be verified by a Krasovski-Barbashin argument.

So optimal control can be used for finding and verifying stabilizing feedbacks.

If the optimal cost $\pi(x)$ and optimal feedback $\kappa(x)$ can be found....

All methods for solving HJB equations suffer from the

Curse of Dimensionality

All methods for solving HJB equations suffer from the

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 at least $m = 3$. The position and attitude control problem for an airplane has state dimension $n = 12$ and control dimension at least $m = 4$.

All methods for solving HJB equations suffer from the

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 at least $m = 3$. 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 nodes 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.

Exception: Linear Quadratic Regulator

If the dynamics is linear and the Lagrangian quadratic

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

then the optimal cost is quadratic and optimal feedback is linear

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

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

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

Exception: Linear Quadratic Regulator

If the dynamics is linear and the Lagrangian quadratic

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

then the optimal cost is quadratic and optimal feedback is linear

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

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 asymptotically stabilizing, i.e., all the poles of $F + GK$ are in the open left half plane.

Al'brekht's Method

Al'brecht developed the power series method for solving the HJB equations for smooth systems that have Taylor series expansions.

$$
\begin{aligned}
f(x,u) &= Fx + Gu + f^{[2]}(x,u) + f^{[3]}(x,u) + \ldots \\
l(x,u) &= \frac{1}{2}\left(x'QX + u'Ru\right) + l^{[3]}(x,u) + l^{[4]}(x,u) + \ldots
\end{aligned}
$$

Al'brekht's Method

Al'brecht developed the power series method for solving the HJB equations for smooth systems that have Taylor series expansions.

$$\begin{aligned} f(x,u) &= Fx + Gu + f^{[2]}(x,u) + f^{[3]}(x,u) + \dots \\ l(x,u) &= \frac{1}{2}\left(x'QX + u'Ru\right) + l^{[3]}(x,u) + l^{[4]}(x,u) + \dots \end{aligned}$$

He assumed that the optimal cost and optimal feedback had similar expansions

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

Al'brekht's Method

Al'brecht developed the power series method for solving the HJB equations for smooth systems that have Taylor series expansions.

$$
\begin{aligned}
f(x,u) &= Fx + Gu + f^{[2]}(x,u) + f^{[3]}(x,u) + \dots \\
l(x,u) &= \frac{1}{2}\left(x'QX + u'Ru\right) + l^{[3]}(x,u) + l^{[4]}(x,u) + \dots
\end{aligned}
$$

He assumed that the optimal cost and optimal feedback had similar expansions

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

He plugged these expansions into HJB. At the lowest degrees he got the familiar LQR equations

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

Al'brekht's Method

Next the unknown degree three terms $\pi^{[3]}(x)$ of the cost and the unknown degree two terms $\kappa^{[2]}(x)$ of the feedback satisfy

$$0 = \frac{\partial \pi^{[3]}}{\partial x}(x)(F + GK)x + x'Pf^{[2]}(x, Kx) + l^{[3]}(x, Kx)$$

$$0 = \frac{\partial \pi^{[3]}}{\partial x}(x)G + x'P\frac{\partial f^{[2]}}{\partial u}(x, Kx) + \frac{\partial^{[3]}}{\partial u}(x, Kx) + (\kappa^{[2]}(x)'R$$

Al'brekht's Method

Next the unknown degree three terms $\pi^{[3]}(x)$ of the cost and the unknown degree two terms $\kappa^{[2]}(x)$ of the feedback satisfy

$$0 = \frac{\partial \pi^{[3]}}{\partial x}(x)(F + GK)x + x'Pf^{[2]}(x, Kx) + l^{[3]}(x, Kx)$$

$$0 = \frac{\partial \pi^{[3]}}{\partial x}(x)G + x'P\frac{\partial f^{[2]}}{\partial u}(x, Kx) + \frac{\partial^{[3]}}{\partial u}(x, Kx) + (\kappa^{[2]}(x)'R$$

Notice the linear triangular structure. Under the standard LQR assumptions the first linear equation is always solvable for $\pi^{[3]}(x)$ because the eigenvalues of the map

$$\pi^{[3]}(x) \mapsto \frac{\partial \pi^{[3]}}{\partial x}(x)(F + GK)x$$

are sums of three eigenvalues of $F + GK$, $\sigma(F + GK) < 0$.

Al'brekht's Method

Next the unknown degree three terms $\pi^{[3]}(x)$ of the cost and the unknown degree two terms $\kappa^{[2]}(x)$ of the feedback satisfy

$$0 = \frac{\partial \pi^{[3]}}{\partial x}(x)(F + GK)x + x'Pf^{[2]}(x, Kx) + l^{[3]}(x, Kx)$$

$$0 = \frac{\partial \pi^{[3]}}{\partial x}(x)G + x'P\frac{\partial f^{[2]}}{\partial u}(x, Kx) + \frac{\partial^{[3]}}{\partial u}(x, Kx) + (\kappa^{[2]}(x)'R$$

Notice the linear triangular structure. Under the standard LQR assumptions the first linear equation is always solvable for $\pi^{[3]}(x)$ because the eigenvalues of the map

$$\pi^{[3]}(x) \mapsto \frac{\partial \pi^{[3]}}{\partial x}(x)(F + GK)x$$

are sums of three eigenvalues of $F + GK$, $\sigma(F + GK) < 0$.

Then the second linear equation is always solvable for $\kappa^{[2]}(x)$ because R is assumed to be invertible.

Al'brekht's Method

The higher degree terms are found in a similar fashion.

Al'brekht's Method

The higher degree terms are found in a similar fashion.

This method has been implemented in the MATLAB based Nonlinear Systems Toolbox to arbitrary degree and dimensions.

Al'brekht's Method

The higher degree terms are found in a similar fashion.

This method has been implemented in the MATLAB based Nonlinear Systems Toolbox to arbitrary degree and dimensions.

Al'brekht's method works in reasonable dimensions. For example, 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 this four year old laptop.

Al'brekht's Method

The higher degree terms are found in a similar fashion.

This method has been implemented in the MATLAB based Nonlinear Systems Toolbox to arbitrary degree and dimensions.

Al'brekht's method works in reasonable dimensions. For example, 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 this four year old laptop.

Al'brekht's method is fast. This laptop took 0.082334 seconds to solve the HJB equations for the satellite attitude problem, $(n = 6,\ m = 3)$, to degree 4 in $\pi(x)$ and degree 3 in $\kappa(x)$.

Pros and Cons of Al'brekht's Method

- **Al'brekht's Method can be extended to discrete time and time varing problems.**

Pros and Cons of Al'brekht's Method

- **Al'brekht's Method can be extended to discrete time and time varing problems.**
- **Except for the Riccati equation it only involves solving linear equations and Matlab software is available.**

Pros and Cons of Al'brekht's Method

- **Al'brekht's Method can be extended to discrete time and time varing problems.**
- **Except for the Riccati equation it only involves solving linear equations and Matlab software is available.**
- **Al'brekht's Method is restricted to smooth systems with no state or control constraints.**

Pros and Cons of Al'brekht's Method

- **Al'brekht's Method can be extended to discrete time and time varing problems.**
- **Except for the Riccati equation it only involves solving linear equations and Matlab software is available.**
- **Al'brekht's Method is restricted to smooth systems with no state or control constraints.**
- **The LQR part must yield a Hurwitz $F + GK$.**

Pros and Cons of Al'brekht's Method

- **Al'brekht's Method can be extended to discrete time and time varing problems.**
- **Except for the Riccati equation it only involves solving linear equations and Matlab software is available.**
- **Al'brekht's Method is restricted to smooth systems with no state or control constraints.**
- **The LQR part must yield a Hurwitz $F + GK$.**
- **The software is fast and it can used for systems of moderately large state dimension, e.g., $n = 25$, $m = 8$.**

Pros and Cons of Al'brekht's Method

- **Al'brekht's Method can be extended to discrete time and time varing problems.**
- **Except for the Riccati equation it only involves solving linear equations and Matlab software is available.**
- **Al'brekht's Method is restricted to smooth systems with no state or control constraints.**
- **The LQR part must yield a Hurwitz $F + GK$.**
- **The software is fast and it 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 Al'brekht's Method

- **Al'brekht's Method can be extended to discrete time and time varing problems.**
- **Except for the Riccati equation it only involves solving linear equations and Matlab software is available.**
- **Al'brekht's Method is restricted to smooth systems with no state or control constraints.**
- **The LQR part must yield a Hurwitz $F + GK$.**
- **The software is fast and it 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 to do so. It can also decrease it.**

Pros and Cons of Al'brekht's Method

- **Al'brekht's Method can be extended to discrete time and time varing problems.**
- **Except for the Riccati equation it only involves solving linear equations and Matlab software is available.**
- **Al'brekht's Method is restricted to smooth systems with no state or control constraints.**
- **The LQR part must yield a Hurwitz $F + GK$.**
- **The software is fast and it 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 to do so. 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 Al'brekht's Method

- **Besides supplying a stabilizing feedback, Al'brekht's Method supplies a candidate Lyapunov function so the basin of attraction can be estimated. (This can take much longer than Al'brekht's method itself.)**

Pros and Cons of Al'brekht's Method

- **Besides supplying a stabilizing feedback, Al'brekht's Method supplies a candidate Lyapunov function so the basin of attraction can be estimated. (This can take much longer than Al'brekht's method itself.)**
- **Because the feedback is a higher degree polynomial it can easily lead to finite escape time in the model and catastrophe in the actual plant.**

Pros and Cons of Al'brekht's Method

- **Besides supplying a stabilizing feedback, Al'brekht's Method supplies a candidate Lyapunov function so the basin of attraction can be estimated. (This can take much longer than Al'brekht's method itself.)**
- **Because the feedback is a higher degree polynomial it can easily lead to finite escape time in the model and catastrophe in the actual plant.**
- **Because the feedback is a higher degree polynomial it can easily violate state and/or control constraints.**

Pros and Cons of Al'brekht's Method

- **Besides supplying a stabilizing feedback, Al'brekht's Method supplies a candidate Lyapunov function so the basin of attraction can be estimated. (This can take much longer than Al'brekht's method itself.)**
- **Because the feedback is a higher degree polynomial it can easily lead to finite escape time in the model and catastrophe in the actual plant.**
- **Because the feedback is a higher degree polynomial it can easily violate state and/or control constraints.**
- **On the other hand higher degree penalty terms can be added to the Lagrangian to enforce such constraints.**

Pros and Cons of Al'brekht's Method

- **Besides supplying a stabilizing feedback, Al'brekht's Method supplies a candidate Lyapunov function so the basin of attraction can be estimated. (This can take much longer than Al'brekht's method itself.)**
- **Because the feedback is a higher degree polynomial it can easily lead to finite escape time in the model and catastrophe in the actual plant.**
- **Because the feedback is a higher degree polynomial it can easily violate state and/or control constraints.**
- **On the other hand higher degree penalty terms can be added to the Lagrangian to enforce such constraints.**
- **Even though it is higher degree, Al'brekht's Method is a local method but patchy extensions are possible.**

Pros and Cons of Al'brekht's Method

- **Besides supplying a stabilizing feedback, Al'brekht's Method supplies a candidate Lyapunov function so the basin of attraction can be estimated. (This can take much longer than Al'brekht's method itself.)**
- **Because the feedback is a higher degree polynomial it can easily lead to finite escape time in the model and catastrophe in the actual plant.**
- **Because the feedback is a higher degree polynomial it can easily violate state and/or control constraints.**
- **On the other hand higher degree penalty terms can be added to the Lagrangian to enforce such constraints.**
- **Even though it is higher degree, Al'brekht's Method is a local method but patchy extensions are possible.**
- **Al'brekht's Method can be used to speed up Model Predictive Control (MPC)!**

Discrete Time Infinite Horizon Optimal Control

Minimize

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

Discrete Time Infinite Horizon Optimal Control

Minimize

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

subject to

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

where $x^+(t) = x(t+1)$.

Dynamic Programming Equations

Optimal Cost $\pi(x)$, **Optimal Feedback** $u = \kappa(x)$.

Dynamic Programming Equations

Optimal Cost $\pi(x)$, **Optimal Feedback** $u = \kappa(x)$.

Bellman's Dynamic Programming Equations

$$\begin{aligned} \pi(x) &= \pi(f(x,\kappa(x)) + l(x,\kappa(x)) \\ \kappa(x) &= \mathrm{argmin}_u \{\pi(f(x,u)) + l(x,u)\} \end{aligned}$$

Al'brekht's Method in Discrete Time

Expand everything in power series and collect terms of lowest degree. This yields the familiar discrete time LQR equations,

$$
\begin{aligned}
P &= F'PF + Q - F'PG\left(R + G'PG\right)^{-1} G'PF \\
K &= -\left(R + G'PG\right)^{-1} G'PF
\end{aligned}
$$

Al'brekht's Method in Discrete Time

Expand everything in power series and collect terms of lowest degree. This yields the familiar discrete time LQR equations,

$$
\begin{aligned}
P &= F'PF + Q - F'PG\left(R + G'PG\right)^{-1} G'PF \\
K &= -\left(R + G'PG\right)^{-1} G'PF
\end{aligned}
$$

At the next degrees we get

$$
\begin{aligned}
0 &= \pi^{[3]}((F+GK)x) - \pi^{[3]}(x) \\
&\quad + l^{[3]}(x,Kx) + (f^{[2]}(x,Kx))'P(F+GK)x
\end{aligned}
$$

$$
\begin{aligned}
0 &= \frac{\partial \pi^{[3]}}{\partial x}((F+GK)x)G + \frac{\partial l^{[3]}}{\partial u}(x,Kx) \\
&\quad + (f^{[2]}(x,Kx)'PG + ((F+GK)x)'P\frac{\partial f^{[2]}}{\partial u}(x,Kx) \\
&\quad + (\kappa^{[2]}(x))'(R+G'PG)
\end{aligned}
$$

Al'brekht's Method in Discrete Time

Again these equations are linear and triangular as only the unknown $\pi^{[3]}$ appears in the first one.

This linear equation is always solvable because the eigenvalues of the map

$$\pi^{[3]}(x) \mapsto \pi^{[3]}(x) - \pi^{[3]}((F + GK)x)$$

are 1 minus the product of three eigenvalues of $F + GK$.

Under the standard discrete LQR assumptions all the eigenvalues of $F + GK$ are strictly inside the unit disk so their triple products are also strictly inside the unit disk.

Al'brekht's Method in Discrete Time

Again these equations are linear and triangular as only the unknown $\pi^{[3]}$ appears in the first one.

This linear equation is always solvable because the eigenvalues of the map

$$\pi^{[3]}(x) \mapsto \pi^{[3]}(x) - \pi^{[3]}((F+GK)x)$$

are 1 minus the product of three eigenvalues of $F+GK$.

Under the standard discrete LQR assumptions all the eigenvalues of $F+GK$ are strictly inside the unit disk so their triple products are also strictly inside the unit disk.

Then the second linear equation is always solvable for the other unknown $\kappa^{[2]}$ because R is assumed to be invertible.

Al'brekht's Method in Discrete Time

Again these equations are linear and triangular as only the unknown $\pi^{[3]}$ appears in the first one.

This linear equation is always solvable because the eigenvalues of the map

$$\pi^{[3]}(x) \mapsto \pi^{[3]}(x) - \pi^{[3]}((F+GK)x)$$

are 1 minus the product of three eigenvalues of $F+GK$.

Under the standard discrete LQR assumptions all the eigenvalues of $F+GK$ are strictly inside the unit disk so their triple products are also strictly inside the unit disk.

Then the second linear equation is always solvable for the other unknown $\kappa^{[2]}$ because R is assumed to be invertible.

The higher degree terms are found in a similar fashion.

Al'brekht with Equality Constraints

We now show how Al'brekht's method can be extended to handle an equality constraint of the form

$$\mathbf{0} = \boldsymbol{\alpha}(x, u)$$

Al'brekht with Equality Constraints

We now show how Al'brekht's method can be extended to handle an equality constraint of the form

$$0 \;=\; \alpha(x,u)$$

We assume that this k dimensional constraint is satisfied at $x(0),\ u(0)$ then for it to continue to be satisfied the differential constraint must hold,

$$0 \;=\; a(x,u) = L_{f(x,u)}\alpha(x,u) = \frac{\partial \alpha}{\partial x}(x,u)f(x,u)$$

Al'brekht with Equality Constraints

We now show how Al'brekht's method can be extended to handle an equality constraint of the form

$$0 = \alpha(x,u)$$

We assume that this k dimensional constraint is satisfied at $x(0),\ u(0)$ then for it to continue to be satisfied the differential constraint must hold,

$$0 = a(x,u) = L_{f(x,u)}\alpha(x,u) = \frac{\partial \alpha}{\partial x}(x,u)f(x,u)$$

Because the constraint must hold at $x = 0,\ u = 0$

$$a(x,u) = Ax + Bu + a^{[2]}(x,u) + a^{[3]}(x,u) + \dots$$

Al'brekht with Equality Constraints

We now show how Al'brekht's method can be extended to handle an equality constraint of the form

$$0 = \alpha(x, u)$$

We assume that this k dimensional constraint is satisfied at $x(0),\ u(0)$ then for it to continue to be satisfied the differential constraint must hold,

$$0 = a(x, u) = L_{f(x,u)}\alpha(x, u) = \frac{\partial \alpha}{\partial x}(x, u) f(x, u)$$

Because the constraint must hold at $x = 0,\ u = 0$

$$a(x, u) = Ax + Bu + a^{[2]}(x, u) + a^{[3]}(x, u) + \dots$$

We assume that $\frac{\partial \alpha}{\partial u}(x, u)$ is of full row rank. If this does not hold then at least one of the constraints can be expressed in terms of x alone. Then we just reduce the state dimension.

Al'brekht with Equality Constraints

We attach this constraint to the second HJB equation with a state dependent Lagrange multiplier $\lambda(x) \in I\!R^{k\times 1}$

$$\kappa(x) = \operatorname{argmin}_{u,\lambda} \left\{ \frac{\partial \pi}{\partial x}(x)(f(x) + g(x)u) + l(x,u) + \lambda'(x)a(x,u) \right\}$$

Al'brekht with Equality Constraints

We attach this constraint to the second HJB equation with a state dependent Lagrange multiplier $\lambda(x) \in \mathbb{R}^{k\times 1}$

$$\kappa(x) = \textbf{argmin}_{u,\lambda} \left\{ \frac{\partial \pi}{\partial x}(x)(f(x) + g(x)u) + l(x,u) + \lambda'(x)a(x,u) \right\}$$

Because this is strictly convex in u and linear in λ it reduces to

$$\begin{aligned} 0 &= \frac{\partial \pi}{\partial x}(x)g(x) + \frac{\partial l}{\partial u}(x,\kappa(x)) + \lambda'(x)\frac{\partial a}{\partial u}(x,\kappa(x)) \\ 0 &= a(x,\kappa(x)) \end{aligned}$$

Al'brekht with Equality Constraints

We attach this constraint to the second HJB equation with a state dependent Lagrange multiplier $\lambda(x) \in \mathbb{R}^{k\times 1}$

$$\kappa(x) = \text{argmin}_{u,\lambda}\left\{\frac{\partial \pi}{\partial x}(x)(f(x)+g(x)u)+l(x,u)+\lambda'(x)a(x,u)\right\}$$

Because this is strictly convex in u and linear in λ it reduces to

$$\begin{aligned} 0 &= \frac{\partial \pi}{\partial x}(x)g(x)+\frac{\partial l}{\partial u}(x,\kappa(x))+\lambda'(x)\frac{\partial a}{\partial u}(x,\kappa(x)) \\ 0 &= a(x,\kappa(x)) \end{aligned}$$

Assume

$$\lambda(x) = Lx+\lambda^{[2]}(x)+\lambda^{[3]}(x)+\dots$$

Al'brekht with Equality Constraints

We attach this constraint to the second HJB equation with a state dependent Lagrange multiplier $\lambda(x) \in I\!R^{k\times 1}$

$$\kappa(x) = \mathrm{argmin}_{u,\lambda}\left\{\frac{\partial \pi}{\partial x}(x)(f(x)+g(x)u)+l(x,u)+\lambda'(x)a(x,u)\right\}$$

Because this is strictly convex in u and linear in λ it reduces to

$$\begin{aligned} 0 &= \frac{\partial \pi}{\partial x}(x)g(x)+\frac{\partial l}{\partial u}(x,\kappa(x))+\lambda'(x)\frac{\partial a}{\partial u}(x,\kappa(x)) \\ 0 &= a(x,\kappa(x)) \end{aligned}$$

Assume

$$\lambda(x) = Lx+\lambda^{[2]}(x)+\lambda^{[3]}(x)+\dots$$

Plug this and the other expansions into the HJB equations and collect terms of lowest degree.

Al'brekht with Equality Constraints

This leads to an unusual Riccati equation for P and L

$$0 = PF + F'P + Q - (PG + S)R^{-1}(G'P + S') + L'BR^{-1}B'L$$

The optimal feedback linear gain is

$$K = -R^{-1}\left(G'P + S' + B'L\right)$$

Al'brekht with Equality Constraints

This leads to an unusual Riccati equation for P and L

$$0 = PF + F'P + Q - (PG+S)R^{-1}(G'P+S') + L'BR^{-1}B'L$$

The optimal feedback linear gain is

$$K = -R^{-1}\left(G'P + S' + B'L\right)$$

Since B has full row rank we can reorder the controls so that the last k columns of B form an invertible $k \times k$ matrix B^2. Partition accordingly

$$\begin{aligned} u &= \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ B &= \begin{bmatrix} B^1 & B^2 \end{bmatrix} \\ G &= \begin{bmatrix} G^1 & G^2 \end{bmatrix} \\ S &= \begin{bmatrix} S^1 & S^2 \end{bmatrix} \\ R &= \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} \end{aligned}$$

Al'brekht with Equality Constraints

The linear part of the constraint forces the partial feedback

$$u_2 \;=\; -(B^2)^{-1}\left(Ax + B^1 u_1\right)$$

This leads to an unconstrained LQR problem in the free control u_1 which is of dimension $m-k$. We solve this problem for P, K_1 and compute L from the linear part of the second HJB equation.

Al'brekht with Equality Constraints

The linear part of the constraint forces the partial feedback

$$u_2 \;=\; -(B^2)^{-1}\left(Ax + B^1 u_1\right)$$

This leads to an unconstrained LQR problem in the free control u_1 which is of dimension $m-k$. We solve this problem for $P,\ K_1$ and compute L from the linear part of the second HJB equation.

The quadratic part of the cost and the linear part of the feedback are

$$\begin{aligned}
\pi^{[2]}(x) &= \frac{1}{2}x'Px \\
\kappa^{[1]}(x) &= \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} x = \begin{bmatrix} K_1 \\ -(B^2)^{-1}\left(A + B^1 K_1\right)x \end{bmatrix}
\end{aligned}$$

Al'brekht with Equality Constraints

The cubic part of the first HJB equation is

$$
\begin{aligned}
0 \;=\;& \frac{\partial \pi^{[3]}}{\partial x}(x)(F+GK)x + x'P\left(f^{[2]}(x,Kx) + G\kappa^{[2]}(x)\right) \\
& + l^{[3]}(x,Kx) + x'K'R\,\kappa^{[2]}(x)
\end{aligned}
$$

with the unknowns in red.

Al'brekht with Equality Constraints

The cubic part of the first HJB equation is

$$\begin{aligned} 0 &= \frac{\partial \pi^{[3]}}{\partial x}(x)(F+GK)x + x'P\left(f^{[2]}(x,Kx) + G\kappa^{[2]}(x)\right) \\ &\quad + l^{[3]}(x,Kx) + x'K'R\,\kappa^{[2]}(x) \end{aligned}$$

with the unknowns in red.

This reduces to an equation for $\pi^{[3]}(x)$ alone,

$$\begin{aligned} 0 &= \frac{\partial \pi^{[3]}}{\partial x}(x)(F+GK)x + x'Pf^{[2]}(x,Kx) \\ &\quad + l^{[3]}(x,Kx) + x'La^{[2]}(x,Kx) \end{aligned}$$

Al'brekht with Equality Constraints

The cubic part of the first HJB equation is

$$\begin{aligned} 0 &= \frac{\partial \pi^{[3]}}{\partial x}(x)(F+GK)x + x'P\left(f^{[2]}(x,Kx)+G\kappa^{[2]}(x)\right) \\ &\quad + l^{[3]}(x,Kx) + x'K'R\,\kappa^{[2]}(x) \end{aligned}$$

with the unknowns in red.

This reduces to an equation for $\pi^{[3]}(x)$ alone,

$$\begin{aligned} 0 &= \frac{\partial \pi^{[3]}}{\partial x}(x)(F+GK)x + x'Pf^{[2]}(x,Kx) \\ &\quad + l^{[3]}(x,Kx) + x'La^{[2]}(x,Kx) \end{aligned}$$

If $F+GK$ is Hurwitz then this equation is uniquely solvable for $\pi^{[3]}(x)$.

Al'brekht with Equality Constraints

The quadratic part of the second HJB equation is

$$\begin{aligned} 0 &= \frac{\partial \pi^{[3]}}{\partial x}(x)G + x'P\frac{\partial f^{[2]}}{\partial u}(x,Kx) + \frac{\partial l^{[3]}}{\partial u}(x,Kx) \\ &\quad + x'L'\frac{\partial a^{[2]}}{\partial u}(x,Kx) + (\kappa^{[2]}(x))'R + (\lambda^{[2]}(x))'B \end{aligned}$$

Al'brekht with Equality Constraints

The quadratic part of the second HJB equation is

$$\begin{aligned} 0 &= \frac{\partial \pi^{[3]}}{\partial x}(x)G + x'P\frac{\partial f^{[2]}}{\partial u}(x,Kx) + \frac{\partial l^{[3]}}{\partial u}(x,Kx) \\ &\quad + x'L'\frac{\partial a^{[2]}}{\partial u}(x,Kx) + (\kappa^{[2]}(x))'R + (\lambda^{[2]}(x))'B \end{aligned}$$

The quadratic part of the constraint is

$$0 = B\kappa^{[2]}(x) + a^{[2]}(x,Kx)$$

Al'brekht with Equality Constraints

The quadratic part of the second HJB equation is

$$\begin{aligned} 0 &= \frac{\partial \pi^{[3]}}{\partial x}(x)G + x'P\frac{\partial f^{[2]}}{\partial u}(x,Kx) + \frac{\partial l^{[3]}}{\partial u}(x,Kx) \\ &\quad + x'L'\frac{\partial a^{[2]}}{\partial u}(x,Kx) + (\kappa^{[2]}(x))'R + (\lambda^{[2]}(x))'B \end{aligned}$$

The quadratic part of the constraint is

$$0 = B\kappa^{[2]}(x) + a^{[2]}(x,Kx)$$

$$\begin{bmatrix} R & B' \\ B & 0 \end{bmatrix} \begin{bmatrix} \kappa^{[2]}(x) \\ \lambda^{[2]}(x) \end{bmatrix} = \text{Known Terms}$$

Because B is assumed to be of full row rank this linear equation is uniquely solvable.

Al'brekht with Equality Constraints

The quadratic part of the second HJB equation is

$$\begin{aligned} 0 &= \frac{\partial \pi^{[3]}}{\partial x}(x)G + x'P\frac{\partial f^{[2]}}{\partial u}(x,Kx) + \frac{\partial l^{[3]}}{\partial u}(x,Kx) \\ &\quad + x'L'\frac{\partial a^{[2]}}{\partial u}(x,Kx) + (\kappa^{[2]}(x))'R + (\lambda^{[2]}(x))'B \end{aligned}$$

The quadratic part of the constraint is

$$0 = B\kappa^{[2]}(x) + a^{[2]}(x,Kx)$$

$$\begin{bmatrix} R & B' \\ B & 0 \end{bmatrix} \begin{bmatrix} \kappa^{[2]}(x) \\ \lambda^{[2]}(x) \end{bmatrix} = \text{Known Terms}$$

Because B is assumed to be of full row rank this linear equation is uniquely solvable.

The higher degree terms are found in a similar way.

Al'brekht with Inequality Constraints

Suppose we have the constraint

$$\mathbf{0} \quad \geq \quad \boldsymbol{\beta}(\boldsymbol{x}, \boldsymbol{u})$$

which we assume is not active at the origin $\boldsymbol{\beta}(\mathbf{0}, \mathbf{0}) < \mathbf{0}$.

Al'brekht with Inequality Constraints

Suppose we have the constraint

$$0 \geq \beta(x, u)$$

which we assume is not active at the origin $\beta(0, 0) < 0$.

Frequently such constraints can be handle by adding penalty terms to the Lagrangian $l(x, u)$.

Al'brekht with Inequality Constraints

Suppose we have the constraint

$$0 \geq \beta(x,u)$$

which we assume is not active at the origin $\beta(0,0) < 0$.

Frequently such constraints can be handle by adding penalty terms to the Lagrangian $l(x,u)$.

Here are two simple examples.

Al'brekht with an Inequality State Constraint

Unstable linear system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

Al'brekht with an Inequality State Constraint

Unstable linear system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

Lagrangian

$$l(x,u) = \frac{1}{2}\left(x_1^2 + x_2^2 + u^2\right)$$

Al'brekht with an Inequality State Constraint

Unstable linear system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

Lagrangian

$$l(x, u) = \frac{1}{2} \left(x_1^2 + x_2^2 + u^2 \right)$$

State Constraint

$$x_1 \leq 0.5$$

Al'brekht with an Inequality State Constraint

Unstable linear system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

Lagrangian

$$l(x,u) = \frac{1}{2}\left(x_1^2 + x_2^2 + u^2\right)$$

State Constraint

$$x_1 \leq 0.5$$

Initial Condition

$$\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0.4 \\ 0.7 \end{bmatrix}$$

Al'brekht with a State Inequality Constraint

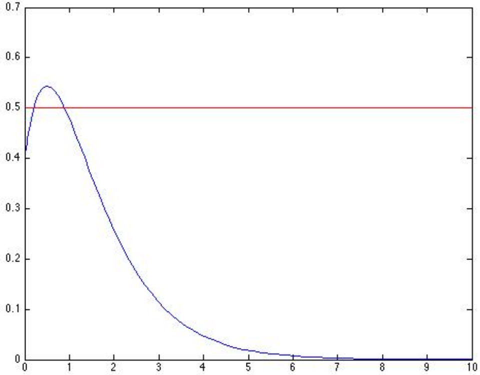
Linear Feedback

$$\begin{aligned} l(x,u) &= \frac{1}{2}\left(x_1^2 + x_2^2 + u^2\right) \\ u &= -2.4142x_1 - 2.4142x_2 \end{aligned}$$

Al'brekht with a State Inequality Constraint

Linear Feedback

$$
\begin{aligned}
l(x,u) &= \frac{1}{2}\left(x_1^2 + x_2^2 + u^2\right) \\
u &= -2.4142x_1 - 2.4142x_2
\end{aligned}
$$



Al'brekht with a State Inequality Constraint

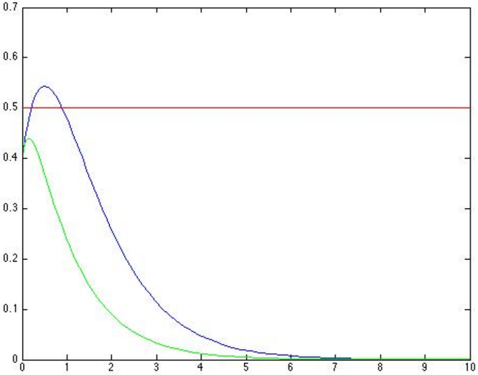
Quintic Feedback

$$
\begin{aligned}
l(x,u) &= \frac{1}{2}\left(x_1^2 + x_2^2 + u^2\right) + 32x_1^5 + 64x_1^6 \\
u &= -2.41x_1 - 2.41x_2 \\
&\quad -22.62x_1^4 - 28.49x_1^3x_2 - 15.36x_1^2x_2^2 - 4.00x_1x_2^3 - 0.41x_2^4 \\
&\quad -45.25x_1^5 - 67.01x_1^4x_2 - 45.60x_1^3x_2^2 - 16.93x_1^2x_2^3 - 3.34x_1x_2^4 - 0.27x_2^5
\end{aligned}
$$

Al'brekht with a State Inequality Constraint

Quintic Feedback

$$
\begin{aligned}
l(x,u) &= \frac{1}{2}\left(x_1^2 + x_2^2 + u^2\right) + 32x_1^5 + 64x_1^6 \\
u &= -2.41x_1 - 2.41x_2 \\
&\quad -22.62x_1^4 - 28.49x_1^3x_2 - 15.36x_1^2x_2^2 - 4.00x_1x_2^3 - 0.41x_2^4 \\
&\quad -45.25x_1^5 - 67.01x_1^4x_2 - 45.60x_1^3x_2^2 - 16.93x_1^2x_2^3 - 3.34x_1x_2^4 - 0.27x_2^5
\end{aligned}
$$



Al'brekht with a Control Inequality Constraint

$$
\begin{aligned}
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\
l(x,u) &= \frac{1}{2}\left(x_1^2 + x_2^2 + u^2\right)
\end{aligned}
$$

Al'brekht with a Control Inequality Constraint

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$l(x, u) = \frac{1}{2}\left(x_1^2 + x_2^2 + u^2\right)$$

Control Constraint

$$|u| \leq 1$$

Al'brekht with a Control Inequality Constraint

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$l(x, u) = \frac{1}{2}\left(x_1^2 + x_2^2 + u^2\right)$$

Control Constraint

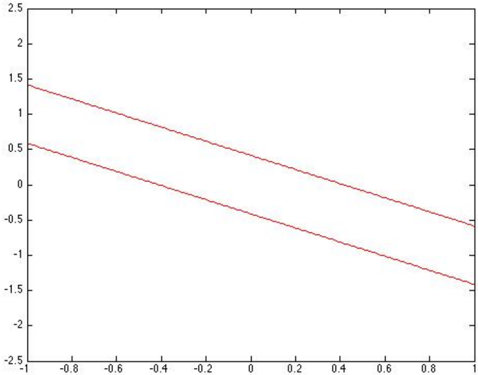
$$|u| \leq 1$$

Linear Feedback

$$u = -2.4142(x_1 + x_2)$$

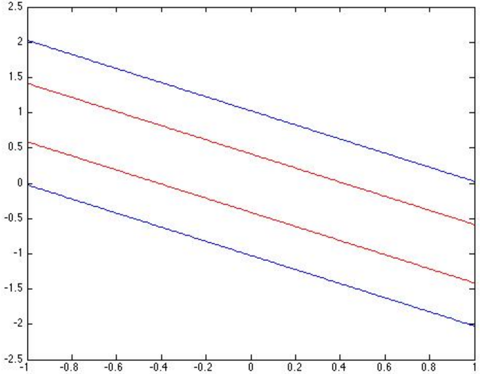
Feasible Region of Linear Feedback

$$u = -2.4142(x_1 + x_2)$$



Feasible Region of Cubic Feedback

$$
\begin{aligned}
l(x,u) &= \frac{1}{2}\left(x_1^2 + x_2^2 + u^2\right) + \frac{1}{10}u^4 \\
u &= -2.4142(x_1 + x_2) - 3.2263(x_1 + x_2)^3
\end{aligned}
$$



Finite Horizon Optimal Control Problem

Minimize

$$\sum_{t=t_0}^{t_f-1} l(t, x(t), u(t)) + \pi_f(x(t_f))$$

subject to

$$\begin{aligned} x^+ &= f(t, x, u) \\ x(t_0) &= x^0 \end{aligned}$$

Finite Horizon Optimal Control Problem

Minimize

$$\sum_{t=t_0}^{t_f-1} l(t, x(t), u(t)) + \pi_f(x(t_f))$$

subject to

$$\begin{aligned} x^+ &= f(t, x, u) \\ x(t_0) &= x^0 \end{aligned}$$

The terminal cost is $\pi_f(x(t_f))$.

Finite Horizon Optimal Control Problem

Minimize

$$\sum_{t=t_0}^{t_f-1} l(t, x(t), u(t)) + \pi_f(x(t_f))$$

subject to

$$\begin{aligned} x^+ &= f(t, x, u) \\ x(t_0) &= x^0 \end{aligned}$$

The terminal cost is $\pi_f(x(t_f))$.

The optimal cost is $\pi(t_0, x^0)$.

Finite Horizon Optimal Control Problem

Minimize

$$\sum_{t=t_0}^{t_f-1} l(t, x(t), u(t)) + \pi_f(x(t_f))$$

subject to

$$\begin{aligned} x^+ &= f(t, x, u) \\ x(t_0) &= x^0 \end{aligned}$$

The terminal cost is $\pi_f(x(t_f))$.

The optimal cost is $\pi(t_0, x^0)$.

The optimal feedback is $u(t_0) = \kappa(t_0, x^0)$.

Finite Horizon Optimal Control Problem

The Dynamic Promming Equations for this problem are

$$\begin{aligned}
\pi(t,x) &= \pi(t+1, f(t,x,\kappa(t,x))) + l(t,x,\kappa(t,x)) \\
\kappa(t,x) &= \operatorname{argmin}_u \{\pi(t+1, f(t,x,u)) + l(t,x,u)\} \\
\pi(t_f,x) &= \pi_f(x)
\end{aligned}$$

Finite Horizon Optimal Control Problem

The Dynamic Promming Equations for this problem are

$$\begin{aligned} \pi(t,x) &= \pi(t+1, f(t,x,\kappa(t,x))) + l(t,x,\kappa(t,x)) \\ \kappa(t,x) &= \text{argmin}_u \{\pi(t+1, f(t,x,u)) + l(t,x,u)\} \\ \pi(t_f,x) &= \pi_f(x) \end{aligned}$$

Assuming the Hamiltonian is strictly convex in u then $\kappa(t,x)$ is the solution of

$$0 = \frac{\partial \pi}{\partial x}(t+1, f(t,x,\kappa(t,x)))\frac{\partial f}{\partial u}(t,x,\kappa(t,x)) + \frac{\partial l}{\partial u}(t,x,\kappa(t,x))$$

Finite Horizon Optimal Control Problem

The Dynamic Promming Equations for this problem are

$$
\begin{aligned}
\pi(t,x) &= \pi(t+1, f(t,x,\kappa(t,x))) + l(t,x,\kappa(t,x)) \\
\kappa(t,x) &= \textbf{argmin}_u \left\{ \pi(t+1, f(t,x,u)) + l(t,x,u) \right\} \\
\pi(t_f,x) &= \pi_f(x)
\end{aligned}
$$

Assuming the Hamiltonian is strictly convex in u then $\kappa(t,x)$ is the solution of

$$
0 = \frac{\partial \pi}{\partial x}(t+1, f(t,x,\kappa(t,x)))\frac{\partial f}{\partial u}(t,x,\kappa(t,x)) + \frac{\partial l}{\partial u}(t,x,\kappa(t,x))
$$

Henceforth we shall assume that f is linear in u and l is quadratic in u which ensures strict convexity,

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

where $R(t,x) > 0$, $\begin{bmatrix} Q(t,x) & S(t,x) \\ S'(t,x) & R(t,x) \end{bmatrix} \geq 0.$

Finite Horizon Optimal Control Problem

If this can be solved for $\kappa(t, x)$ then the result is plugged into the first DPE. It becomes a difference equation for $\pi(t, x)$ that is solved backward in time from the final condition $\pi(t_f, x) = \pi_f(x)$.

Finite Horizon Optimal Control Problem

If this can be solved for $\kappa(t,x)$ then the result is plugged into the first DPE. It becomes a difference equation for $\pi(t,x)$ that is solved backward in time from the final condition $\pi(t_f,x) = \pi_f(x)$.

Of course this is easier said than done!

Al'brekht's Method around an Optimal Trajectory

Let $x^*(t),\ u^*(t)$ be an optimal trajectory and define variational coordinates

$$\begin{aligned} z &= x - x^*(t) \\ v &= u - u^*(t) \end{aligned}$$

Al'brekht's Method around an Optimal Trajectory

Let $x^*(t),\ u^*(t)$ be an optimal trajectory and define variational coordinates

$$\begin{aligned} z &= x - x^*(t) \\ v &= u - u^*(t) \end{aligned}$$

In these coordinates the variational dynamics and the variational Lagrangian are given by

$$\begin{aligned} \tilde{f}(t,z,v) &= f(t,x^*(t)+z,u^*(t)+v) - f(t,x^*(t),u^*(t)) \\ \tilde{l}(t,z,v) &= l(t,x^*(t)+z,u^*(t)+v) - l(t,x^*(t),u^*(t)) \end{aligned}$$

Al'brekht's Method around an Optimal Trajectory

Let $x^*(t),\ u^*(t)$ be an optimal trajectory and define variational coordinates

$$
\begin{aligned}
z &= x - x^*(t) \\
v &= u - u^*(t)
\end{aligned}
$$

In these coordinates the variational dynamics and the variational Lagrangian are given by

$$
\begin{aligned}
\tilde{f}(t, z, v) &= f(t, x^*(t) + z, u^*(t) + v) - f(t, x^*(t), u^*(t)) \\
\tilde{l}(t, z, v) &= l(t, x^*(t) + z, u^*(t) + v) - l(t, x^*(t), u^*(t))
\end{aligned}
$$

The optimal variational cost and optimal variational feedback are given by

$$
\begin{aligned}
\tilde{\pi}(t_0, z^0) &= \pi(t, x^*(t) + z) - \pi(t, x^*(t)) \\
v(t_0) &= \tilde{\kappa}(t_0, z^0) = \kappa(t_0, x^*(t_0) + z^0) - u^*(t)
\end{aligned}
$$

Al'brekht's Method around an Optimal Trajectory

Following Al'brekht we expand everthing in power series in $z,\ v$

$$
\begin{aligned}
\tilde{f}(t,z) &= \tilde{F}(t)z + \tilde{f}^{[2]}(t,z) + \dots \\
\tilde{g}(t,z)v &= \tilde{G}(t)v + \tilde{g}^{[1]}(t,z)v + \dots \\
\tilde{l}(t,z,v) &= \tilde{\lambda}(t)z + \tilde{\mu}(t)v + \frac{1}{2}\left(z'\tilde{Q}(t)z + 2z'\tilde{S}(t)v + v'\tilde{R}(t)v\right) \\
&\quad + \tilde{l}^{[3]}(t,z,v) + \dots \\
\tilde{\pi}_f(z) &= \tilde{\rho}_f z + \frac{1}{2}z'\tilde{P}_f z + \tilde{\pi}_f^{[3]}(z) + \dots \\
\tilde{\pi}(t,z) &= \tilde{\rho}(t)z + \frac{1}{2}z'\tilde{P}(t)z + \tilde{\pi}^{[3]}(t,z) + \dots \\
\tilde{\kappa}(t,z) &= \tilde{K}(t)z + \tilde{\kappa}^{[2]}(t,z) + \dots
\end{aligned}
$$

Al'brekht's Method around an Optimal Trajectory

Following Al'brekht we expand everthing in power series in $z, \ v$

$$
\begin{aligned}
\tilde{f}(t,z) &= \tilde{F}(t)z + \tilde{f}^{[2]}(t,z) + \dots \\
\tilde{g}(t,z)v &= \tilde{G}(t)v + \tilde{g}^{[1]}(t,z)v + \dots \\
\tilde{l}(t,z,v) &= \tilde{\lambda}(t)z + \tilde{\mu}(t)v + \frac{1}{2}\left(z'\tilde{Q}(t)z + 2z'\tilde{S}(t)v + v'\tilde{R}(t)v\right) \\
&\quad + \tilde{l}^{[3]}(t,z,v) + \dots \\
\tilde{\pi}_f(z) &= \tilde{\rho}_f z + \frac{1}{2}z'\tilde{P}_f z + \tilde{\pi}_f^{[3]}(z) + \dots \\
\tilde{\pi}(t,z) &= \tilde{\rho}(t)z + \frac{1}{2}z'\tilde{P}(t)z + \tilde{\pi}^{[3]}(t,z) + \dots \\
\tilde{\kappa}(t,z) &= \tilde{K}(t)z + \tilde{\kappa}^{[2]}(t,z) + \dots
\end{aligned}
$$

What is different is the presence of linear terms in $\tilde{l}(t,z,v), \ \tilde{\pi}_f(z), \ \tilde{\pi}(t,z)$.

Al'brekht's Method around an Optimal Trajectory

Variational Dynamic Programming Equations (VDPE)

Al'brekht's Method around an Optimal Trajectory

Variational Dynamic Programming Equations (VDPE)

VDPE 1

$$\tilde{\pi}(t,z) \;=\; \tilde{\pi}(t+1,\tilde{f}(t,z,\tilde{\kappa}(t,z))) + \tilde{l}(t,z,\tilde{\kappa}(t,z))$$

VDPE 2

$$0 = \frac{\partial \tilde{\pi}}{\partial x}(t+1,f(t,x,\tilde{\kappa}(t,x)))\frac{\partial \tilde{f}}{\partial v}(t,x,\tilde{\kappa}(t,x)) + \frac{\partial \tilde{l}}{\partial v}(t,x,\tilde{\kappa}(t,x))$$

VDPE 3

$$\tilde{\pi}(t_f,x) \;=\; \tilde{\pi}_f(x)$$

Al'brekht's Method around an Optimal Trajectory

Variational Dynamic Programming Equations (VDPE)

VDPE 1

$$\tilde{\pi}(t,z) \;=\; \tilde{\pi}(t+1,\tilde{f}(t,z,\tilde{\kappa}(t,z))) + \tilde{l}(t,z,\tilde{\kappa}(t,z))$$

VDPE 2

$$0 = \frac{\partial \tilde{\pi}}{\partial x}(t+1,f(t,x,\tilde{\kappa}(t,x)))\frac{\partial \tilde{f}}{\partial v}(t,x,\tilde{\kappa}(t,x)) + \frac{\partial \tilde{l}}{\partial v}(t,x,\tilde{\kappa}(t,x))$$

VDPE 3

$$\tilde{\pi}(t_f,x) \;=\; \tilde{\pi}_f(x)$$

We plug the above expansions into these equations and start collecting terms of lowest degree.

Al'brekht's Method around an Optimal Trajectory

VDPE 1, Degree 0

$$\mathbf{0} = \mathbf{0}$$

VDPE 2, Degree 0

$$\mathbf{0} = \tilde{\mu}(t) + \tilde{\rho}(t+1)G(t)$$

VDPE 1, Degree 1

$$\tilde{\rho}(t) = \tilde{\lambda}(t) + \tilde{\rho}(t+1)\tilde{F}(t)$$

VDPE 2, Degree 1

$$\mathbf{0} = z'\tilde{S}(t) + z'\tilde{P}(t+1)\tilde{G}(t) + \tilde{\rho}(t+1)\tilde{g}^{[1]}(t,z) + z'\tilde{K}'(t)\tilde{R}(t)$$

Al'brekht's Method around an Optimal Trajectory

VDPE 1, Degree 2

$$\begin{aligned} z'\tilde{P}(t)z &= \tilde{\mu}(t)\tilde{\kappa}^{[2]}(t,z) + (\tilde{\kappa}^{[2]}(t,z))'\tilde{\mu}'(t) \\ &\quad + z'\tilde{Q}(t)z + z'\tilde{S}(t)\tilde{K}(t)z + z'\tilde{K}'(t)\tilde{S}'(t)z \\ &\quad + z'\tilde{K}'(t)\tilde{R}'(t)\tilde{K}(t)z \\ &\quad + z'(\tilde{F}(t) + \tilde{G}(t)\tilde{K}(t))'\tilde{P}(t+1)(\tilde{F}(t) + \tilde{G}(t)\tilde{K}(t))z \end{aligned}$$

VDPE 2, Degree 2

$$\begin{aligned} 0 &= \frac{\partial \tilde{l}^{[3]}}{\partial v}(t,z,\tilde{K}(t,z)) \\ &\quad + \tilde{\rho}(t+1))\tilde{g}^{[2]}(t+1,(\tilde{F}(t) + \tilde{G}(t)\tilde{K}(t))z) \\ &\quad + z'(\tilde{F}(t) + \tilde{G}(t)\tilde{K}(t))'\tilde{P}(t+1)\tilde{g}^{[1]}(t+1,(\tilde{F}(t) \\ &\quad + \tilde{G}(t)\tilde{K}(t))z) + \tilde{G}(t)\tilde{K}(t))z) + \frac{\partial \tilde{\pi}^{[3]}}{\partial z}(t+1,(\tilde{F}(t) + \tilde{G}(t)\tilde{K} \\ &\quad + (\tilde{\kappa}^{[2]}(t,z)))'\tilde{R}^{(}t) + \frac{\partial \tilde{\pi}^{[3]}}{\partial z}(t+1,(\tilde{F}(t) + \tilde{G}(t)\tilde{K}(t))z)\tilde{G} \\ &\quad + (\tilde{\kappa}^{[2]}(t,z)))'\tilde{R}(t) \end{aligned}$$

Al'brekht's Method around an Optimal Trajectory

There are four equations in the four unknowns in $\tilde{\rho}(t), \tilde{P}(t), \tilde{K}(t), \tilde{\kappa}^{[2]}(t,z)$ at time t .

$$
\begin{aligned}
\tilde{\rho}(t) &= \tilde{\lambda}(t) + \tilde{\rho}(t+1)\tilde{F}(t) \\
0 &= z'\tilde{S}(t) + z'\tilde{P}(t+1)\tilde{G}(t) + \tilde{\rho}(t+1)\tilde{g}^{[1]}(t,z) \\
&\quad +\tilde{K}'(t)\tilde{R}(t) \\
z'\tilde{P}(t)z &= \tilde{\mu}(t)\tilde{\kappa}^{[2]}(t,z) + (\tilde{\kappa}^{[2]}(t,z))'\tilde{\mu}'(t) \\
&\quad +z'\tilde{Q}(t)z + z'\tilde{S}(t)\tilde{K}(t)z + z'\tilde{K}'(t)\tilde{S}'(t)z \\
&\quad +z'\tilde{K}'(t)\tilde{R}'(t)\tilde{K}(t)z \\
0 &= \frac{\partial \tilde{l}^{[3]}}{\partial v}(t,z,\tilde{K}(t,z)) \\
&\quad +\tilde{\rho}(t+1))\tilde{g}^{[2]}(t+1,(\tilde{F}(t)+\tilde{G}(t)\tilde{K}(t))z) \\
&\quad +z'(\tilde{F}(t)+\tilde{G}(t)\tilde{K}(t))'\tilde{P}(t+1)\tilde{g}^{[1]}(t+1,(\tilde{F}(t) \\
&\quad +\tilde{G}(t)\tilde{K}(t))z) + \frac{\partial \tilde{\pi}^{[3]}}{\partial z}(t+1,(\tilde{F}(t)+\tilde{G}(t)\tilde{K}(t))z)\tilde{G} \\
&\quad +(\tilde{\kappa}^{[2]}(t,z)))'\tilde{R}^{(}t)
\end{aligned}
$$

Al'brekht's Method around an Optimal Trajectory

These reduce to a Riccati difference equation for $\tilde{P}(t)$ and three linear difference equations for $\tilde{\rho}(t),\ \tilde{K}(t),\ \tilde{\kappa}^{[2]}(t,z)$

These equations run backward in time from the terminal conditions

$$
\begin{aligned}
\tilde{\rho}(t_f) &= \tilde{\rho}_f \\
\tilde{P}(t_f) &= \tilde{P}_f \\
\tilde{K}(t_f) &= \tilde{K}_f \\
\tilde{\kappa}^{[2]}(t,z) &= \tilde{\kappa}^{[2]}_f(z)
\end{aligned}
$$

Al'brekht's Method around an Optimal Trajectory

These reduce to a Riccati difference equation for $\tilde{P}(t)$ and three linear difference equations for $\tilde{\rho}(t),\ \tilde{K}(t),\ \tilde{\kappa}^{[2]}(t,z)$

These equations run backward in time from the terminal conditions

$$
\begin{aligned}
\tilde{\rho}(t_f) &= \tilde{\rho}_f \\
\tilde{P}(t_f) &= \tilde{P}_f \\
\tilde{K}(t_f) &= \tilde{K}_f \\
\tilde{\kappa}^{[2]}(t,z) &= \tilde{\kappa}^{[2]}_f(z)
\end{aligned}
$$

Notice that we need the terminal cost $\tilde{\kappa}_f(z)$ as well as the terminal cost $\tilde{\pi}_f(z)$.

Al'brekht's Method around an Optimal Trajectory

At the next degree we get more difference equations

$$\tilde{\pi}^{[3]}(t,z) = \tilde{l}^{[3]}(t,z,\tilde{K}(t)) + z'\tilde{S}(t)\tilde{\kappa}^{[2]}(t,z) + (\tilde{\kappa}^{[2]}(t,z))'\tilde{S}'(t)z$$
$$+\tilde{\pi}^{[3]}(t+1,z) + z'(\tilde{F}(t)+\tilde{G}(t)\tilde{K}(t))'\tilde{P}(t+1)\tilde{f}^{[2]}(t,z)$$
$$+\tilde{g}^{[1]}(t,z)\tilde{K}(t)z + \tilde{G}(t)\tilde{\kappa}^{[2]}(t,z)$$
$$+\rho(t+1)(\tilde{f}^{[3]}(t,z) + \tilde{g}^{[2]}(t,z)\tilde{K}(t)z$$
$$+\tilde{g}^{[1]}(t,z)\tilde{\kappa}^{[2]}(t,z) + \tilde{G}(t)\tilde{\kappa}^{[3]}(t,z)$$

$$0 = \frac{\partial l^{[4]}}{\partial v}(t,z,\tilde{K}(t)z) + \left(\frac{\partial l^{[3]}}{\partial v}(t,z,\tilde{\kappa}^{[2]}(t,z)\right)^{[3]}$$
$$+\tilde{\rho}(t+1)\tilde{g}^{[3]}(t,z) + z'(\tilde{F}(t)+\tilde{G}(t)\tilde{K}(t))'\tilde{P}(t+1)\tilde{g}^{[2]}(t,z)$$
$$+\frac{\partial\tilde{\pi}^{[3]}}{\partial z}(t+1,z)\tilde{g}^{[1]}(t,z) + \frac{\partial\tilde{\pi}^{[4]}}{\partial z}(t+1,z)\tilde{G}(t) + (\tilde{\kappa}^{[3]}(t,z))'\tilde{R}(t)$$

Al'brekht's Method around an Optimal Trajectory

At the next degree we get more difference equations

$$\tilde{\pi}^{[3]}(t,z) = \tilde{l}^{[3]}(t,z,\tilde{K}(t)) + z'\tilde{S}(t)\tilde{\kappa}^{[2]}(t,z) + (\tilde{\kappa}^{[2]}(t,z))'\tilde{S}'(t)z$$
$$+\tilde{\pi}^{[3]}(t+1,z) + z'(\tilde{F}(t) + \tilde{G}(t)\tilde{K}(t))'\tilde{P}(t+1)\tilde{f}^{[2]}(t,z)$$
$$+\tilde{g}^{[1]}(t,z)\tilde{K}(t)z + \tilde{G}(t)\tilde{\kappa}^{[2]}(t,z)$$
$$+\rho(t+1)(\tilde{f}^{[3]}(t,z) + \tilde{g}^{[2]}(t,z)\tilde{K}(t)z$$
$$+\tilde{g}^{[1]}(t,z)\tilde{\kappa}^{[2]}(t,z) + \tilde{G}(t)\tilde{\kappa}^{[3]}(t,z)$$

$$0 = \frac{\partial l^{[4]}}{\partial v}(t,z,\tilde{K}(t)z) + \left(\frac{\partial l^{[3]}}{\partial v}(t,z,\tilde{\kappa}^{[2]}(t,z)\right)^{[3]}$$
$$+\tilde{\rho}(t+1)\tilde{g}^{[3]}(t,z) + z'(\tilde{F}(t) + \tilde{G}(t)\tilde{K}(t))'\tilde{P}(t+1)\tilde{g}^{[2]}(t,z)$$
$$+\frac{\partial \tilde{\pi}^{[3]}}{\partial z}(t+1,z)\tilde{g}^{[1]}(t,z) + \frac{\partial \tilde{\pi}^{[4]}}{\partial z}(t+1,z)\tilde{G}(t) + (\tilde{\kappa}^{[3]}(t,z))'\tilde{R}(t)$$

Notice the linear triangular structure and the presence of $\tilde{\pi}^{[4]}(t+1,z)$ in the second equation. If we stop at degree three then this is set to zero.

Al'brekht's Method around an Optimal Trajectory

At the next degree we get more difference equations

$$\begin{aligned}
\tilde{\pi}^{[3]}(t,z) &= \tilde{l}^{[3]}(t,z,\tilde{K}(t)) + z'\tilde{S}(t)\tilde{\kappa}^{[2]}(t,z) + (\tilde{\kappa}^{[2]}(t,z))'\tilde{S}'(t)z \\
&+\tilde{\pi}^{[3]}(t+1,z) + z'(\tilde{F}(t)+\tilde{G}(t)\tilde{K}(t))'\tilde{P}(t+1)\tilde{f}^{[2]}(t,z) \\
&+\tilde{g}^{[1]}(t,z)\tilde{K}(t)z + \tilde{G}(t)\tilde{\kappa}^{[2]}(t,z) \\
&+\rho(t+1)(\tilde{f}^{[3]}(t,z) + \tilde{g}^{[2]}(t,z)\tilde{K}(t)z \\
&+\tilde{g}^{[1]}(t,z)\tilde{\kappa}^{[2]}(t,z) + \tilde{G}(t)\tilde{\kappa}^{[3]}(t,z)
\end{aligned}$$

$$\begin{aligned}
0 &= \frac{\partial l^{[4]}}{\partial v}(t,z,\tilde{K}(t)z) + \left(\frac{\partial l^{[3]}}{\partial v}(t,z,\tilde{\kappa}^{[2]}(t,z)\right)^{[3]} \\
&+\tilde{\rho}(t+1)\tilde{g}^{[3]}(t,z) + z'(\tilde{F}(t)+\tilde{G}(t)\tilde{K}(t))'\tilde{P}(t+1)\tilde{g}^{[2]}(t,z) \\
&+\frac{\partial\tilde{\pi}^{[3]}}{\partial z}(t+1,z)\tilde{g}^{[1]}(t,z) + \frac{\partial\tilde{\pi}^{[4]}}{\partial z}(t+1,z)\tilde{G}(t) + (\tilde{\kappa}^{[3]}(t,z))'\tilde{R}(t)
\end{aligned}$$

Notice the linear triangular structure and the presence of $\tilde{\pi}^{[4]}(t+1,z)$ in the second equation. If we stop at degree three then this is set to zero.

The higher degree terms are found in a similar fashion.

Patchy Method

The domain of stability of a polynomial solution to the HJB equations is the domain where the polynomial cost is a valid Lyapunov function for the closed loop dynamics using the polynomial feedback.

Patchy Method

The domain of stability of a polynomial solution to the HJB equations is the domain where the polynomial cost is a valid Lyapunov function for the closed loop dynamics using the polynomial feedback.

Taking Al'brekht to higher degree makes the computed solution more accurate but does not necessarily increase the domain of stability of the solution.

Patchy Method

The domain of stability of a polynomial solution to the HJB equations is the domain where the polynomial cost is a valid Lyapunov function for the closed loop dynamics using the polynomial feedback.

Taking Al'brekht to higher degree makes the computed solution more accurate but does not necessarily increase the domain of stability of the solution.

Moreover using high degree stability can lead to instability and finite escape times.

Patchy Method

The domain of stability of a polynomial solution to the HJB equations is the domain where the polynomial cost is a valid Lyapunov function for the closed loop dynamics using the polynomial feedback.

Taking Al'brekht to higher degree makes the computed solution more accurate but does not necessarily increase the domain of stability of the solution.

Moreover using high degree stability can lead to instability and finite escape times.

Therefore Navasca, Hunt, Aguilar and Krener developed the patchy method.

Patchy Method

The domain of stability of a polynomial solution to the HJB equations is the domain where the polynomial cost is a valid Lyapunov function for the closed loop dynamics using the polynomial feedback.

Taking Al'brekht to higher degree makes the computed solution more accurate but does not necessarily increase the domain of stability of the solution.

Moreover using high degree stability can lead to instability and finite escape times.

Therefore Navasca, Hunt, Aguilar and Krener developed the patchy method.

Compute the Al'brekht solution and accept it on some sublevel set of the computed optimal cost.

Patchy Method

The domain of stability of a polynomial solution to the HJB equations is the domain where the polynomial cost is a valid Lyapunov function for the closed loop dynamics using the polynomial feedback.

Taking Al'brekht to higher degree makes the computed solution more accurate but does not necessarily increase the domain of stability of the solution.

Moreover using high degree stability can lead to instability and finite escape times.

Therefore Navasca, Hunt, Aguilar and Krener developed the patchy method.

Compute the Al'brekht solution and accept it on some sublevel set of the computed optimal cost.

Then go to a point on the boundary of the sublevel set and recompute the power series. Accept this solution on a patch.

Patchy Method

The domain of stability of a polynomial solution to the HJB equations is the domain where the polynomial cost is a valid Lyapunov function for the closed loop dynamics using the polynomial feedback.

Taking Al'brekht to higher degree makes the computed solution more accurate but does not necessarily increase the domain of stability of the solution.

Moreover using high degree stability can lead to instability and finite escape times.

Therefore Navasca, Hunt, Aguilar and Krener developed the patchy method.

Compute the Al'brekht solution and accept it on some sublevel set of the computed optimal cost.

Then go to a point on the boundary of the sublevel set and recompute the power series. Accept this solution on a patch.

Patchy Methods

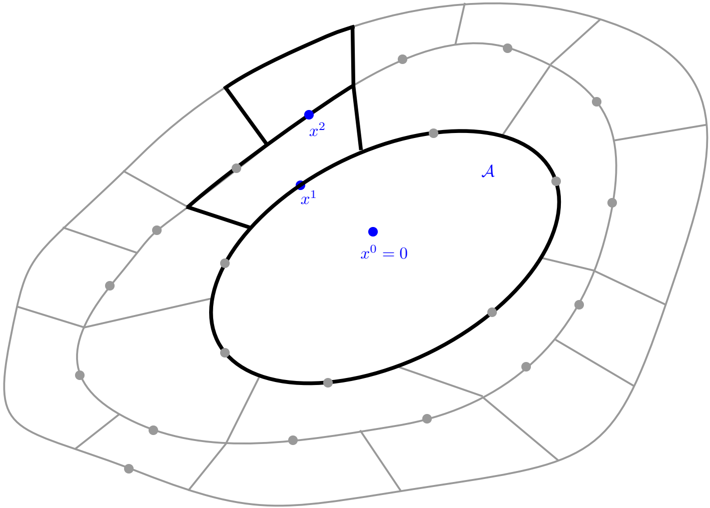


Figure : Sequence of Patches

Patch Calculation

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

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

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

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

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

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

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 the partial derivatives of $\pi(x)$ on the previous patch.

Patch Calculation

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

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

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 the partial derivatives of $\pi(x)$ on the previous patch.

If we assume that $\frac{\partial \pi^1}{\partial x}(x^1) = z \frac{\partial \pi^0}{\partial x}(x^1)$ then at degree one 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

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

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 the partial derivatives of $\pi(x)$ on the previous patch.

If we assume that $\frac{\partial \pi^1}{\partial x}(x^1) = z \frac{\partial \pi^0}{\partial x}(x^1)$ then at degree one 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.

Patchy Methods

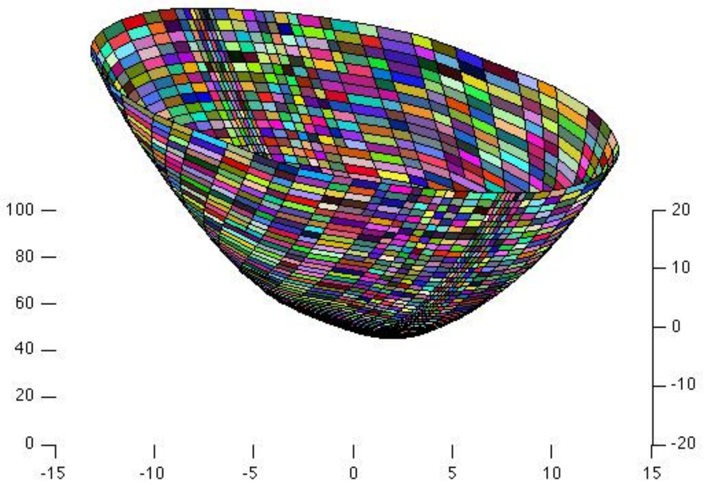


Figure : Optimal Cost of Inverting a Pendulum by a Torque at its Axis

Invert a Pendulum

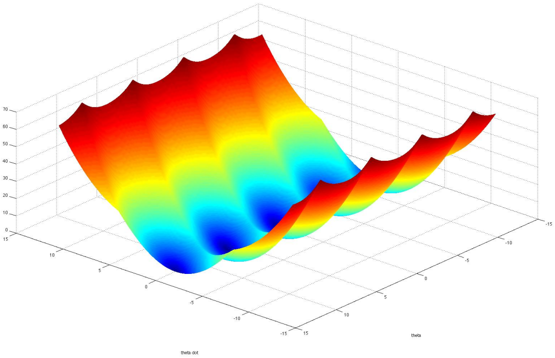


Figure : Periodicity of the Optimal Cost

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.

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.

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 $d+1$.

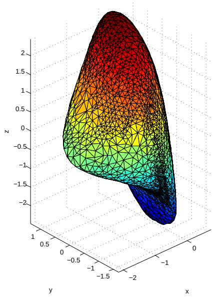
	Max Error	Max Rel Error	Error Factor
LQR $(d=1)$	0.3543	0.8860	54.56
Al'brecht $(d=3)$	0.1636	0.4101	25.16
Patchy $(d=3)$	0.0065	0.0239	1

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

The patchy method can also be used when there are constraints. The constraint may be active or inactive on a patch.

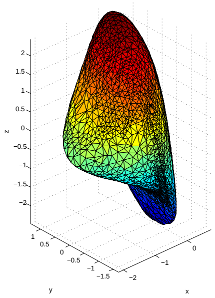
Three Dimensional Example

Patchy method applied to a three dimensional problem



Three Dimensional Example

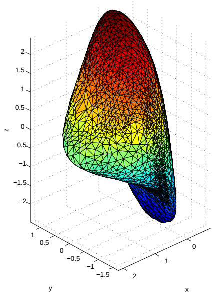
Patchy method applied to a three dimensional problem



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

Three Dimensional Example

Patchy method applied to a three dimensional problem



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

But adding one or two shells of patches to Al'brekht is feasible in moderate dimensions. The feedback can be linear on these shells which reduces the possibility of finite escape by the closed loop dynamics.

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 $n = 3$.

Except for Al'brekht 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 $n = 3$.

Except for Al'brekht it is questionable whether any of these methods are feasible when $n = 4$ or $n = 5$.

Except for Al'brekht 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 $n = 3$.

Except for Al'brekht it is questionable whether any of these methods are feasible when $n = 4$ or $n = 5$.

Except for Al'brekht it is unlikely that any of these methods are feasible when $n \geq 6$.

But Al'brekht only yields a local solution about the origin.

Conclusions

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

Except for Al'brekht it is questionable whether any of these methods are feasible when $n = 4$ or $n = 5$.

Except for Al'brekht it is unlikely that any of these methods are feasible when $n \geq 6$.

But Al'brekht only yields a local solution about the origin.

Al'brekht can be enlarged slightly by one or two shells of patches.

Conclusions

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

Except for Al'brekht it is questionable whether any of these methods are feasible when $n = 4$ or $n = 5$.

Except for Al'brekht it is unlikely that any of these methods are feasible when $n \geq 6$.

But Al'brekht only yields a local solution about the origin.

Al'brekht can be enlarged slightly by one or two shells of patches.

To make the solution global we combine Al'brekht with Model Predictive Control (MPC).

Model Predictive Control

Consider the infinite horizon problem of minimizing

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

subject to

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

Model Predictive Control

Minimization over the infinite horizon is too difficult so we 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 we 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

$$\sum_{t=0}^{T-1} l(x(t), u(t)) + \pi_T(x(T))$$

subject to

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

Model Predictive Control

Minimization over the infinite horizon is too difficult so we 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

$$\sum_{t=0}^{T-1} l(x(t), u(t)) + \pi_T(x(T))$$

subject to

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

The decision variables are $u(0), \ldots, u(T-1)$.

Model Predictive Control

Then pass this nonlinear program to a fast solver to find the optimal $u^0(0), \ldots, u^0(T-1)$. This needs to be done in less than the time step.

Model Predictive Control

Then pass this nonlinear program to a fast solver to find the optimal $u^0(0), \dots, u^0(T-1)$. This needs to be done in less than the time step.

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

Model Predictive Control

Then pass this nonlinear program to a fast solver to find the optimal $u^0(0), \ldots, u^0(T-1)$. This needs to be done in less than the time step.

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

Then between times 1 and 2 solve the problem of minimizing

$$\sum_{t=1}^{T} l(x(t), u(t)) + \pi_T(x(T+1))$$

subject to

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

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

Model Predictive Control

Then pass this nonlinear program to a fast solver to find the optimal $u^0(0), \ldots, u^0(T-1)$. This needs to be done in less than the time step.

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

Then between times 1 and 2 solve the problem of minimizing

$$\sum_{t=1}^{T} l(x(t), u(t)) + \pi_T(x(T+1))$$

subject to

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

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

Use the control $u^1(1)$ to get the state to $x^2 = x(2)$, 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 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 must be short enough to accurately approximate it.**
- **The time step should be long enough so that the nonlinear program can be solved in one time step. Actually it needs to be solved in a small fraction of a time step so that we can employ $u(t)$ nearly at time 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 must be short enough to accurately approximate it.**
- **The time step should be long enough so that the nonlinear program can be solved in one time step. Actually it needs to be solved in a small fraction of a time step so that we can employ $u(t)$ nearly at time t .**
- **The horizon T must be short enough so that the nonlinear program can be solved in a small fraction of a 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 must be short enough to accurately approximate it.**
- **The time step should be long enough so that the nonlinear program can be solved in one time step. Actually it needs to be solved in a small fraction of a time step so that we can employ $u(t)$ nearly at time t .**
- **The horizon T must be short enough so that the nonlinear program can be solved in a small fraction of a time step.**
- **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 must be short enough to accurately approximate it.**
- **The time step should be long enough so that the nonlinear program can be solved in one time step. Actually it needs to be solved in a small fraction of a time step so that we can employ $u(t)$ nearly at time t .**
- **The horizon T must be short enough so that the nonlinear program can be solved in a small fraction of a time step.**
- **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 $u^0(0), \dots, u^0(T-1)$ 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 $u^0(1), \ldots, u^0(T-1)$ as the initial guess for $u^1(1), \ldots, u^1(T-1)$.**

Model Predictive Control

- **This is not as much a problem with later initial guesses because we can take $u^0(1), \dots, u^0(T-1)$ as the initial guess for $u^1(1), \dots, u^1(T-1)$.**
- **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 $u^0(1), \ldots, u^0(T-1)$ as the initial guess for $u^1(1), \ldots, u^1(T-1)$.**
- **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 $\tilde{x}^0(T)$ is the T^{th} state generated by the last control sequence. $u^0(0), \ldots, u^0(T-1)$**

Model Predictive Control

- This is not as much a problem with later initial guesses because we can take $u^0(1), \ldots, u^0(T-1)$ as the initial guess for $u^1(1), \ldots, u^1(T-1)$.
- 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 $\tilde{x}^0(T)$ is the T^{th} state generated by the last control sequence. $u^0(0), \ldots, u^0(T-1)$
- Al'brekht alone or with a shell or two of patches can furnish $\pi_T(x)$ and $\kappa_T(x)$ on a reasonably large $\mathcal{X}_T$!

Speeding Up Model Predictive Control

Currently MPC is only used for relatively slow and reasonably stable processes such as chemical process control.

Speeding Up Model Predictive Control

Currently MPC is only used for relatively slow and reasonably stable processes such as chemical process control.

There is a need to develop extensions of MPC that can be used for fast processes such as aircraft.

Speeding Up Model Predictive Control

Currently MPC is only used for relatively slow and reasonably stable processes such as chemical process control.

There is a need to develop extensions of MPC that can be used for fast processes such as aircraft.

Using Al'brekht alone or with a shell or two of patches partially achieves that goal by enlarging the terminal set $\mathcal{X}_T$ so that the horizon T can be shortened thereby simplifying the on-line optimization.

Speeding Up Model Predictive Control

Currently MPC is only used for relatively slow and reasonably stable processes such as chemical process control.

There is a need to develop extensions of MPC that can be used for fast processes such as aircraft.

Using Al'brekht alone or with a shell or two of patches partially achieves that goal by enlarging the terminal set $\mathcal{X}_T$ so that the horizon T can be shortened thereby simplifying the on-line optimization.

Al'brekht around an optimal trajectory also can be used to increase the available computational time which we now explain.

Speeding Up Model Predictive Control

Standard MPC waits until $x(t)$ is known to compute $u^t(t), \ldots, u^t(t+T-1)$.

Speeding Up Model Predictive Control

Standard MPC waits until $x(t)$ is known to compute $u^t(t), \ldots, u^t(t+T-1)$.

This needs to be done in a small fraction of a time step so that the control $u(t) = u^t(t)$ can be employed for the rest of the time step.

Speeding Up Model Predictive Control

Standard MPC waits until $x(t)$ is known to compute $u^t(t), \ldots, u^t(t+T-1)$.

This needs to be done in a small fraction of a time step so that the control $u(t) = u^t(t)$ can be employed for the rest of the time step.

Suppose instead we start computing $u^t(t), \ldots, u^t(t+T-1)$ at time $t-1$ based on the model predicted value $\hat{x}(t) = f(x(t-1, u(t-1))$ of $x(t)$.

Speeding Up Model Predictive Control

Standard MPC waits until $x(t)$ is known to compute $u^t(t), \dots, u^t(t+T-1)$.

This needs to be done in a small fraction of a time step so that the control $u(t) = u^t(t)$ can be employed for the rest of the time step.

Suppose instead we start computing $u^t(t), \dots, u^t(t+T-1)$ at time $t-1$ based on the model predicted value $\hat{x}(t) = f(x(t-1, u(t-1))$ of $x(t)$.

The actual value of $x(t)$ will probably be different from its predicted value of $\hat{x}(t)$ but probably not that different.

Speeding Up Model Predictive Control

Standard MPC waits until $x(t)$ is known to compute $u^t(t), \ldots, u^t(t+T-1)$.

This needs to be done in a small fraction of a time step so that the control $u(t) = u^t(t)$ can be employed for the rest of the time step.

Suppose instead we start computing $u^t(t), \ldots, u^t(t+T-1)$ at time $t-1$ based on the model predicted value $\hat{x}(t) = f(x(t-1, u(t-1))$ of $x(t)$.

The actual value of $x(t)$ will probably be different from its predicted value of $\hat{x}(t)$ but probably not that different.

So we compute the variational $\tilde{\pi}(s,z),\ \tilde{\kappa}(s,z)$ for $s = t, \ldots, t+T-1$ by Al'brekht around the optimal trajectory generated by $u^t(t), \ldots, u^t(t+T-1)$ and then at time t when $x(t)$ becomes known we use the control $u(t) = u^t(t) + \tilde{\kappa}(t, x(t) - \hat{x}(t))$.

Concluding Remarks

- **Al'brekht's method is the only viable way to solve HJB or DP equations on a reasonably large domain in moderate or large state dimensions.**

Concluding Remarks

- **Al'brekht's method is the only viable way to solve HJB or DP equations on a reasonably large domain in moderate or large state dimensions.**
- **Al'brekht's method can be extended to handle equality and/or inequality constraints.**

Concluding Remarks

- **Al'brekht's method is the only viable way to solve HJB or DP equations on a reasonably large domain in moderate or large state dimensions.**
- **Al'brekht's method can be extended to handle equality and/or inequality constraints.**
- **Patchy extensions can be added to the Al'brekht solution to enlarge it slightly and mitigate the possibility finite escape time in the closed loop system.**

Concluding Remarks

- **Al'brekht's method is the only viable way to solve HJB or DP equations on a reasonably large domain in moderate or large state dimensions.**
- **Al'brekht's method can be extended to handle equality and/or inequality constraints.**
- **Patchy extensions can be added to the Al'brekht solution to enlarge it slightly and mitigate the possibility finite escape time in the closed loop system.**
- **Model Predictive Control with an Al'brekht terminal cost and terminal feedback is a viable approach to solving optimal control problems in moderate dimensions with moderately fast dynamics.**

Concluding Remarks

- **Al'brekht's method is the only viable way to solve HJB or DP equations on a reasonably large domain in moderate or large state dimensions.**
- **Al'brekht's method can be extended to handle equality and/or inequality constraints.**
- **Patchy extensions can be added to the Al'brekht solution to enlarge it slightly and mitigate the possibility finite escape time in the closed loop system.**
- **Model Predictive Control with an Al'brekht terminal cost and terminal feedback is a viable approach to solving optimal control problems in moderate dimensions with moderately fast dynamics.**
- **Model Predictive Estimation and Al'brekht?**

Concluding Remarks

- **Al'brekht's method is the only viable way to solve HJB or DP equations on a reasonably large domain in moderate or large state dimensions.**
- **Al'brekht's method can be extended to handle equality and/or inequality constraints.**
- **Patchy extensions can be added to the Al'brekht solution to enlarge it slightly and mitigate the possibility finite escape time in the closed loop system.**
- **Model Predictive Control with an Al'brekht terminal cost and terminal feedback is a viable approach to solving optimal control problems in moderate dimensions with moderately fast dynamics.**
- **Model Predictive Estimation and Al'brekht?**
- **Al'brekht and patchy with constraints?**

Concluding Remarks

- **Al'brekht's method is the only viable way to solve HJB or DP equations on a reasonably large domain in moderate or large state dimensions.**
- **Al'brekht's method can be extended to handle equality and/or inequality constraints.**
- **Patchy extensions can be added to the Al'brekht solution to enlarge it slightly and mitigate the possibility finite escape time in the closed loop system.**
- **Model Predictive Control with an Al'brekht terminal cost and terminal feedback is a viable approach to solving optimal control problems in moderate dimensions with moderately fast dynamics.**
- **Model Predictive Estimation and Al'brekht?**
- **Al'brekht and patchy with constraints?**
- **For a copy of these slides contact** `ajkrener@ucdavis.edu`