Sensitivity Analysis for LQR
or
How I Spent My Summer Vacation

Philip D Loewen

UBCV Math

30 November 2024

Contents

Table of Contents

QP Basics

Consider $\quad G(z) = \frac{1}{2} z^T \mathbf{Q} z + \mathbf{q}^T z + \gamma, \quad \text{with } \mathbf{Q} = \mathbf{Q}^T.$

For any base point z and offset v,

$$\begin{aligned} G(z+v) &= \tfrac{1}{2}(z+v)^T \mathbf{Q}(z+v) + \mathbf{q}^T(z+v) + \gamma \\ &= \tfrac{1}{2} z^T \mathbf{Q} z + \tfrac{1}{2} v^T \mathbf{Q} z + \tfrac{1}{2} z^T \mathbf{Q} v + \tfrac{1}{2} v^T \mathbf{Q} v + \mathbf{q}^T z + \mathbf{q}^T v + \gamma \\ &= G(z) + \tfrac{1}{2} v^T \mathbf{Q} v + (\mathbf{Q} z + \mathbf{q}) \bullet v. \end{aligned}$$

This reveals $\nabla G(z) = \mathbf{Q} z + \mathbf{q}$. If $\widehat{z}$ is a critical point, i.e., $\nabla G(\widehat{z}) = 0$,

$$G(\widehat{z} + v) = G(\widehat{z}) + \tfrac{1}{2} v^T \mathbf{Q} v.$$

Taking $\quad \widehat{z} = -\mathbf{Q}^{-1}\mathbf{q} \quad$ gives

$$G(\widehat{z}) = \tfrac{1}{2}\left(\mathbf{Q}^{-1}\mathbf{q}\right)^T \mathbf{Q}\left(\mathbf{Q}^{-1}\mathbf{q}\right) - \mathbf{q}^T\left(\mathbf{Q}^{-1}\mathbf{q}\right) + \gamma = -\tfrac{1}{2}\mathbf{q}^T \mathbf{Q}^{-1}\mathbf{q} + \gamma.$$

Perturbed QP

Imagine $z = (x, u)$, with "decision variable" u and "parameter" x:

$$G(x,u) = \frac{1}{2}\begin{bmatrix} x^T & u^T \end{bmatrix}\begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix}\begin{bmatrix} x \\ u \end{bmatrix} + \begin{bmatrix} \mathbf{q}_1^T & \mathbf{q}_2^T \end{bmatrix}\begin{bmatrix} x \\ u \end{bmatrix} + \gamma.$$

Rearrangement surfaces the quadratic u-dependence:

$$G(x,u) = \frac{1}{2}u^T\mathbf{Q}_{22}u + (\mathbf{q}_2 + \mathbf{Q}_{21}x)^T u + \left(\frac{1}{2}x^T\mathbf{Q}_{11}x + \mathbf{q}_1^T x + \gamma\right).$$

Assume $\mathbf{Q}_{22} > 0$. Then $G(x,\cdot)$ is minimized at its CP,

$$\widehat{u}(x) = -\mathbf{Q}_{22}^{-1}(\mathbf{q}_2 + \mathbf{Q}_{21}x) = Kx + k, \quad \text{with} \quad K = -\mathbf{Q}_{22}^{-1}\mathbf{Q}_{21},\ k = -\mathbf{Q}_{22}^{-1}\mathbf{q}_2.$$

The minimum value is a quadratic function of the parameter x:

$$\widehat{G}(x) = G(x,\widehat{u}(x)) = \frac{1}{2}x^T\left[\mathbf{Q}_{11} - K^T\mathbf{Q}_{22}K\right]x + (\mathbf{q}_1 + \mathbf{Q}_{12}k)^T x + \frac{1}{2}\mathbf{q}_2^T k + \gamma.$$

Table of Contents

Standard-Form Linear Quadratic Regulator (LQR)

States x in discrete-time dynamical system start with $x_0 = x_{\text{init}}$ and obey

$$x_{t+1} = Ax_t + Bu_t, \qquad t = 0, 1, \ldots.$$

Problem: Given a step-count T, choose controls $u_0, \ldots, u_{T-1}$ to minimize

$$J = \tfrac{1}{2} x_T^T Q_T x_T + \tfrac{1}{2} \sum_{t=0}^{T-1} \left(x_t^T Q x_t + u_t^T R u_t \right).$$

Condense the notation and generalize slightly:

- stack $z = \begin{bmatrix} x \\ u \end{bmatrix}$; invent matrix $P = [I \quad 0]$ so that $x = Pz$;
- condense the linear dynamics and quadratic cost into block matrices: use F for the dynamics, C for the cost;
- allow lower-order terms in the cost, drift in the dynamics, and t-dependence.

Extended-Form LQR

Recalling $z = \begin{bmatrix} x \\ u \end{bmatrix}$, $P = [I \quad 0]$ so that $Pz = x$,

$$\begin{aligned} \text{min} \quad & J \ = \sum_{t=0}^{T} \left(\tfrac{1}{2} z_t^T C_t z_t + c_t^T z_t \right) \\ \text{over} \quad & u_0, \ldots, u_T, \\ \text{s.t.} \quad & P z_{t+1} = F_t z_t + f_t, \qquad t = 0, \ldots, T-1, \\ & x_0 = P z_0 = x_{\text{init}}. \end{aligned}$$

Special restrictions: $C_0 \sim \begin{bmatrix} 0 & 0 \\ 0 & * \end{bmatrix}$, $c_0 \sim \begin{bmatrix} 0 & * \end{bmatrix}$, $C_T \sim \begin{bmatrix} * & 0 \\ 0 & 0 \end{bmatrix}$, $c_T \sim \begin{bmatrix} * & 0 \end{bmatrix}$.

Extended-Form LQR

Recalling $z = \begin{bmatrix} x \\ u \end{bmatrix}$, $P = [I \quad 0]$ so that $Pz = x$, set up for Dynamic Programming:

- Consider cost-to-go as a scalar-valued function of initial state;

$$
\begin{aligned}
V\ (x) = \min \quad & J \ = \sum_{t=0}^{T} \left(\tfrac{1}{2} z_t^T C_t z_t + c_t^T z_t \right) \\
\text{over} \quad & u_0, \ldots, u_T, \\
\text{s.t.} \quad & P z_{t+1} = F_t z_t + f_t, \qquad t = 0, \ldots, T-1, \\
& x_0 = P z_0 = x.
\end{aligned}
$$

Special restrictions: $C_0 \sim \begin{bmatrix} 0 & 0 \\ 0 & * \end{bmatrix}$, $c_0 \sim \begin{bmatrix} 0 & * \end{bmatrix}$, $C_T \sim \begin{bmatrix} * & 0 \\ 0 & 0 \end{bmatrix}$, $c_T \sim \begin{bmatrix} * & 0 \end{bmatrix}$.

Extended-Form LQR

Recalling $z = \begin{bmatrix} x \\ u \end{bmatrix}$, $P = [I \quad 0]$ so that $Pz = x$, set up for Dynamic Programming:

- Consider cost-to-go as a scalar-valued function of initial state;
- Consider cost-to-go from any start time.

$$\begin{aligned} V_r(x) = \min \quad & J_r = \sum_{t=r}^{T} \left(\tfrac{1}{2} z_t^T C_t z_t + c_t^T z_t \right) \\ \text{over} \quad & u_r, \ldots, u_T, \\ \text{s.t.} \quad & P z_{t+1} = F_t z_t + f_t, \qquad t = r, \ldots, T-1, \\ & x_r = P z_r = x. \end{aligned}$$

- Function V_T is essentially given. Find prior V_r to solve original problem.

Special restrictions: $C_0 \sim \begin{bmatrix} 0 & 0 \\ 0 & * \end{bmatrix}$, $c_0 \sim \begin{bmatrix} 0 & * \end{bmatrix}$, $C_T \sim \begin{bmatrix} * & 0 \\ 0 & 0 \end{bmatrix}$, $c_T \sim \begin{bmatrix} * & 0 \end{bmatrix}$.

Backward Recursion

Principle of Optimality: From state x at time t, best cost-to-go will satisfy

$$V_t(x) = \min_u G_t(x,u), \qquad \text{where } u \sim u_t, \text{ and}$$
$$G_t(z) = \left(\tfrac{1}{2} z^T C_t z + c_t^T z\right) + V_{t+1}\left(F_t z + f_t\right). \qquad \text{Recall } z = (x,u).$$

If V_{t+1} is a quadratic, then G_t will be quadratic also, and so will V_t.
We know V_T is quadratic, so this holds for all V_t, $t = T, \ldots, 0$. So predict

$$V_t(x) = \tfrac{1}{2} x^T \mathbf{V}_t x + \mathbf{v}_t^T x + \beta_t,$$
$$G_t(x,u) = \tfrac{1}{2} z^T \mathbf{Q}_t z + \mathbf{q}_t^T z + \gamma_t, \qquad z = \begin{bmatrix} x \\ u \end{bmatrix}.$$

We have formulas for the minimizer $\widehat{u_t}(x)$ and the minimum value $V_t(x) = \widehat{G}_t(x) = G_t(x, \widehat{u_t}(x))$! Align them with the forms above …

Two-pass solution

Backward: Knowing function V_T gives $\mathbf{V}_T = C_{T,11}$, $\mathbf{v}_T = c_{T,1}$, $\beta_T = 0$.
Counting down with $t = T-1, \ldots, 2, 1, 0$, define, in order,

$$\begin{aligned}
\mathbf{Q}_t &= C_t + F_t^T \mathbf{V}_{t+1} F_t \\
\mathbf{q}_t &= c_t + F_t^T \mathbf{v}_{t+1} + F_t^T \mathbf{V}_{t+1} f_t \\
K_t &= -(\mathbf{Q}_{t,22})^{-1} \mathbf{Q}_{t,21}, \\
k_t &= -(\mathbf{Q}_{t,22})^{-1} \mathbf{q}_{t,2}, \\
\beta_t &= \beta_{t+1} + \tfrac{1}{2} f_t^T \mathbf{V}_{t+1} f_t + \mathbf{v}_{t+1}^T f_t + \tfrac{1}{2} \mathbf{q}_{t,2}^T k_t, \\
\mathbf{v}_t &= \mathbf{q}_{t,1} + \mathbf{Q}_{t,12} k_t, \\
\mathbf{V}_t &= \mathbf{Q}_{t,11} - K_t^T \mathbf{Q}_{t,22} K_t.
\end{aligned}$$

Forward: Given any initial state x_0, proceed with $t = 0, 1, \ldots, T-1$ in

$$u_t = K_t x_t + k_t, \qquad z_t = \begin{bmatrix} x_t \\ u_t \end{bmatrix}, \qquad x_{t+1} = P(F_t z_t + f_t).$$

Table of Contents

Lagrange Multiplier Alternative

We seek $z_0, z_1, \ldots, z_T$ to

$$\text{minimize} \quad J = \sum_{t=0}^{T} \left(\tfrac{1}{2} z_t^T C_t z_t + c_t^T z_t \right) \quad \text{s.t.} \quad P z_{t+1} = F_t z_t + f_t, \quad t = 0, \ldots, T-1.$$

To solve this "all at once", introduce multipliers $\lambda_0, \ldots, \lambda_{T-1}$ and form

$$\mathscr{L} = \sum_{t=0}^{T} \left(\tfrac{1}{2} z_t^T C_t z_t + c_t^T z_t \right) + \sum_{t=0}^{T-1} \lambda_t^T \left(F_t z_t + f_t - P z_{t+1} \right).$$

The KKT system is a collection of linear equations relating z_t, λ_t; typically,

$$\begin{aligned} 0 &= \nabla_{z_t} \mathscr{L} = C_t z_t + c_t + F_t^T \lambda_t - P^T \lambda_{t-1}, \\ 0 &= \nabla_{\lambda_t} \mathscr{L} = F_t z_t + f_t - P z_{t+1}. \end{aligned}$$

In matrix form, ...

Lagrange Multiplier Alternative

… the KKT system involves a *symmetric, block tridiagonal* coefficient matrix:

$$\begin{bmatrix} \ddots & \vdots & \vdots & \vdots & & & \\ \cdots & 0 & -P & 0 & & & \\ \cdots & -P^T & C_t & F_t^T & & & \\ \cdots & 0 & F_t & 0 & -P & 0 & \cdots \\ & & & -P^T & C_{t+1} & F_{t+1}^T & \cdots \\ & & & 0 & F_{t+1} & 0 & \cdots \\ & & & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \lambda_{t-1} \\ z_t \\ \lambda_t \\ z_{t+1} \\ \lambda_{t+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ -f_{t-1} \\ -c_t \\ -f_t \\ -c_{t+1} \\ -f_{t+1} \\ \vdots \end{bmatrix}$$

(Typical 3×3 blocks overlap on their diagonal corners; details of top and bottom block-rows differ slightly.)

$$\begin{aligned} 0 = \nabla_{z_t}\mathscr{L} \quad &\iff \quad -P^T\lambda_{t-1} + C_t z_t + F_t^T\lambda_t &&= -c_t, \\ 0 = \nabla_{\lambda_t}\mathscr{L} \quad &\iff \quad F_t z_t - P z_{t+1} &&= -f_t. \end{aligned}$$

Notes: (1) Symmetry is useful later.
(2) Sparsity and structure can be exploited.

Table of Contents

Gradients

Setup: $(\mathscr{X}, \bullet)$ is a real inner-product space; $f \colon \mathscr{X} \to \mathbb{R}$; $x_0 \in \mathscr{X}$.

Idea: $\nabla f(x_0)$ is the element of $\mathscr{X}$ for which

$$f(x) \approx f(x_0) + \nabla f(x_0) \bullet (x - x_0), \qquad \text{when } x \approx x_0.$$

Calculation: Assert for arbitrary $v \in \mathscr{X}$,

$$\nabla f(x_0) \bullet v = \lim_{t \to 0} \frac{f(x_0 + tv) - f(x_0)}{t} = \frac{d}{dt} f(x_0 + tv)\Big|_{t=0}$$

Examples

Warmup: $\mathscr{X} = \mathbb{R}^{q\times d}$; $X \bullet Y = \sum_{i,j} X_{ij}Y_{ij} = \sum_i \left(XY^T\right)_{ii} = \mathrm{tr}\left(XY^T\right)$

- Function $f(X) = u^T X w$ is linear, so $\nabla f(X) = uw^T$ is a constant matrix with the same shape as X.
- Take $q = d$ and consider $f(X) = u^T X^{-1} w$. This time, for arbitrary $V \in \mathbb{R}^{d\times d}$,

$$\begin{aligned}
\nabla f(X) \bullet V &= \lim_{t\to 0} t^{-1}u^T\left[(X+tV)^{-1} - X^{-1}\right]w \\
&= \lim_{t\to 0} t^{-1}u^T\left[(I+tX^{-1}V)^{-1} - I\right]X^{-1}w \\
&= -u^T X^{-1} V X^{-1} w \qquad \left(\text{use } (I-R)^{-1} = I + R + R^2 + \dots\right) \\
&= -(X^{-T}u)^T(X^{-1}w) \bullet V.
\end{aligned}$$

Result: If X^{-1} exists, $\nabla f(X) = -\left(X^{-T}u\right)\left(X^{-1}w\right)^T = -X^{-T}uw^TX^{-T}$.

Solution Sensitivity in Linear Equations, $Ay = b$

For $A \in \mathbb{R}^{d\times d}$, $b \in \mathbb{R}^d$, and some given "loss function" $\ell \colon \mathbb{R}^d \to \mathbb{R}$, let

$$W(A,b) = \ell\left(A^{-1}b\right) = \ell(y), \qquad \text{where} \quad Ay = b.$$

For $A' \approx A$, $b' \approx b$, and y' from $A'y' = b'$,

$$\begin{aligned} W(A',b') - W(A,b) &= \ell(y') - \ell(y) \\ &\approx \nabla\ell(y) \bullet (y' - y) = \nabla\ell(y)^T \left((A')^{-1}b' - A^{-1}b\right) \\ &\approx \nabla\ell(y)^T \left((A')^{-1} - A^{-1}\right) b + \nabla\ell(y)^T (A')^{-1}(b' - b). \end{aligned}$$

Reconcile this with

$$W(A',b') - W(A,b) \approx \nabla_A W(A,b) \bullet (A' - A) + \nabla_b W(A,b) \bullet (b' - b).$$

Result: $\quad \nabla_b W(A,b) = A^{-T}\nabla\ell(y), \qquad \nabla_A W(A,b) = -A^{-T}\nabla\ell(y)b^T A^{-T}.$

Notice factor $\quad p \triangleq A^{-T}\nabla\ell(y) \quad$ in both expressions; recall $\quad y = A^{-1}b.$

Solution Sensitivity in Linear Equations, $Ay = b$

For $A \in \mathbb{R}^{d\times d}$, $b \in \mathbb{R}^d$, and some given "loss function" $\ell\colon \mathbb{R}^d \to \mathbb{R}$, let

$$W(A,b) = \ell\big(A^{-1}b\big) = \ell(y), \qquad \text{where} \quad Ay = b.$$

To find gradients of W at a point (A,b) where A^{-1} exists,

1. Solve for y: $\quad Ay = b$.
2. Solve for p: $\quad A^T p = \nabla\ell(y)$. $\qquad$ (That's "the adjoint system".)
3. Report: $\quad \nabla_A W(A,b) = -py^T, \qquad \nabla_b W(A,b) = p$.

Notes:

- Calculations for step (1) (e.g., factorizations for matrix A) may be useful in step (2). Especially when $A^T = A$.
- Gradients reported in (3) match element-by-element with the shapes of the inputs A and b for function W.

Special Structure

Now let $W = W(M, m, G, g) = \ell(x)$, where ℓ is given, and x comes from

$$(*) \qquad \begin{bmatrix} M & G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} m \\ g \end{bmatrix}.$$

Assume $M = M^T$ in $\mathbb{R}^{d\times d}$ and $G \in \mathbb{R}^{q\times d}$ both have full rank. Taking

$$A = \begin{bmatrix} M & G^T \\ G & 0 \end{bmatrix}, \qquad y = \begin{bmatrix} x \\ \lambda \end{bmatrix}, \qquad b = \begin{bmatrix} m \\ g \end{bmatrix},$$

we find gradients of W like this:

Special Structure

Now let $W = W(M, m, G, g) = \ell(x)$, where ℓ is given, and x comes from

$$(*) \qquad \begin{bmatrix} M & G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} m \\ g \end{bmatrix}.$$

Assume $M = M^T$ in $\mathbb{R}^{d \times d}$ and $G \in \mathbb{R}^{q \times d}$ both have full rank. Taking

$$A = \begin{bmatrix} M & G^T \\ G & 0 \end{bmatrix}, \qquad y = \begin{bmatrix} x \\ \lambda \end{bmatrix}, \qquad b = \begin{bmatrix} m \\ g \end{bmatrix},$$

we find gradients of W like this:

1. Solve for (x, λ): $\quad (*)$.

Special Structure

Now let $W = W(M, m, G, g) = \ell(x)$, where ℓ is given, and x comes from

$$(*) \qquad \begin{bmatrix} M & G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} m \\ g \end{bmatrix}.$$

Assume $M = M^T$ in $\mathbb{R}^{d\times d}$ and $G \in \mathbb{R}^{q\times d}$ both have full rank. Taking

$$A = \begin{bmatrix} M & G^T \\ G & 0 \end{bmatrix}, \qquad y = \begin{bmatrix} x \\ \lambda \end{bmatrix}, \qquad b = \begin{bmatrix} m \\ g \end{bmatrix},$$

we find gradients of W like this:

1. Solve for (x, λ): $(*)$.
2. Solve for (p_0, p_1): $\begin{bmatrix} M & G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \end{bmatrix} = \begin{bmatrix} \nabla\ell(x) \\ 0 \end{bmatrix}$.

Special Structure

Now let $\quad W = W(M, m, G, g) = \ell(x), \quad$ where ℓ is given, and x comes from

$$(*) \qquad \begin{bmatrix} M & G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} m \\ g \end{bmatrix}.$$

Assume $M = M^T$ in $\mathbb{R}^{d\times d}$ and $G \in \mathbb{R}^{q\times d}$ both have full rank. Taking

$$A = \begin{bmatrix} M & G^T \\ G & 0 \end{bmatrix}, \qquad y = \begin{bmatrix} x \\ \lambda \end{bmatrix}, \qquad b = \begin{bmatrix} m \\ g \end{bmatrix},$$

we find gradients of W like this:

1. Solve for (x, λ): $\quad (*)$.
2. Solve for (p_0, p_1): $\begin{bmatrix} M & G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \end{bmatrix} = \begin{bmatrix} \nabla\ell(x) \\ 0 \end{bmatrix}$. Notice $py^T = \begin{bmatrix} p_0 x^T & p_0 \lambda^T \\ p_1 x^T & p_1 \lambda^T \end{bmatrix}$.

Special Structure

Now let $W = W(M, m, G, g) = \ell(x)$, where ℓ is given, and x comes from

$$(*) \qquad \begin{bmatrix} M & G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} m \\ g \end{bmatrix}.$$

Assume $M = M^T$ in $\mathbb{R}^{d\times d}$ and $G \in \mathbb{R}^{q\times d}$ both have full rank. Taking

$$A = \begin{bmatrix} M & G^T \\ G & 0 \end{bmatrix}, \qquad y = \begin{bmatrix} x \\ \lambda \end{bmatrix}, \qquad b = \begin{bmatrix} m \\ g \end{bmatrix},$$

we find gradients of W like this:

1. Solve for (x, λ): $(*)$.
2. Solve for (p_0, p_1): $\begin{bmatrix} M & G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \end{bmatrix} = \begin{bmatrix} \nabla\ell(x) \\ 0 \end{bmatrix}$. Notice $py^T = \begin{bmatrix} p_0 x^T & p_0 \lambda^T \\ p_1 x^T & p_1 \lambda^T \end{bmatrix}$.
3. Report: $\nabla_m W = p_0, \qquad \nabla_M W = -\frac{1}{2}(p_0 x^T + x p_0^T),$
   $\nabla_g W = p_1, \qquad \nabla_G W = -(p_1 x^T + \lambda p_0^T).$

Why Bother?

Recognize the previous system

$$(*) \qquad \begin{bmatrix} M & G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} m \\ g \end{bmatrix}$$

as the KKT setup for this Quadratic Programming problem:

$$\min_{x \in \mathbb{R}^d} \tfrac{1}{2} x^T M x - m^T x \quad \text{s.t.} \quad Gx = g.$$

(Check: $\mathscr{L} = \frac{1}{2} x^T M x - m \bullet x + G^T \lambda \bullet x - \lambda \bullet g$.)

Idea: Elements M, m, G, g define a QP. The minimizer x depends on them:

$$x = \widehat{x}(M, m, G, g).$$

The steps above make derivatives of $W \triangleq \ell \circ \widehat{x}$, a "simple" case for the Chain rule, fully concrete.

Why Bother?

Recognize the previous system

$$(*) \qquad \begin{bmatrix} M & G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} m \\ g \end{bmatrix}$$

as the KKT setup for this Quadratic Programming problem:

$$\min_{x \in \mathbb{R}^d} \ \tfrac{1}{2} x^T M x - m^T x \quad \text{s.t.} \quad Gx = g.$$

(Check: $\mathscr{L} = \frac{1}{2} x^T M x - m \bullet x + G^T \lambda \bullet x - \lambda \bullet g$.)

Idea: Elements M, m, G, g define a QP. The minimizer x depends on them:

$$x = \widehat{x}(M, m, G, g).$$

The steps above make derivatives of $W \triangleq \ell \circ \widehat{x}$, a “simple” case for the Chain rule, fully concrete. Here’s Step 1 in the painful alternative we’re skipping:

$$\widehat{x} = M^{-1} \left(I - G^T (G M^{-1} G^T)^{-1} G M^{-1} \right) m + M^{-1} G^T \left(G M^{-1} G^T \right)^{-1} g.$$

Table of Contents

Grand Unification

Recall the LQ problem, where $z = (x, u)$, and $x_0 = x_{\text{init}}$ is given:

$$\begin{aligned} \min \quad & J = \sum_{t=0}^{T} \left(\tfrac{1}{2} z_t^T C_t z_t + c_t^T z_t \right) \\ \text{s.t.} \quad & x_{t+1} = F_t z_t + f_t, \qquad t = 0, \ldots, T-1. \end{aligned}$$

The optimal control/state sequence $\widehat{z}$ depends on the coefficients, defining

$$\widehat{z}_t = \widehat{z}_t(F, f, C, c), \qquad t = 0, 1, \ldots, T.$$

Given any smooth function $\ell = \ell(z)$, consider $W = \ell \circ \widehat{z}$, i.e.,

$$W(F, f, C, c) = \ell(\widehat{z}(F, f, C, c)).$$

The KKT interpretation gives immediate access to gradients of W!

Grand Unification

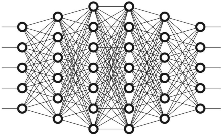
To differentiate $W(F,f,C,c) = \ell(\widehat{z}(F,f,C,c))$ at (F,f,C,c), where

$$\widehat{z} \triangleq \arg\min \quad J = \sum_{t=0}^{T} \left(\tfrac{1}{2} z_t^T C_t z_t + c_t^T z_t\right)$$
$$\text{s.t.} \quad x_{t+1} = F_t z_t + f_t, \qquad t = 0, \ldots, T-1,$$

1. Run the 2-pass method to find $\widehat{z}$ in the nominal problem above.
2. Find the corresponding Lagrange multipliers, $\widehat{\lambda}$. Stack with $\widehat{z}$ to get a vector q solving the KKT system.
3. Calculate $\nabla\ell(\widehat{z})$.
4. Rerun the 2-pass method, after changing $f \to 0$, $c \to -\nabla\ell(\widehat{z})$. (Save work: re-use the matrices in the forward pass.) Name the solution vector p.
5. Extract $\nabla_c W$ and $\nabla_f W$ directly from p.
6. Imagine the matrix $-(pq^T + qp^T)$:
   - find $\nabla_{F_t} W$ in the blocks occupied by F_t in KKT matrix;
   - find $\nabla_{C_t} W$ in the blocks occupied by C_t in KKT matrix.

Table of Contents

Applications



- Direct: LQR is useful. Sensitivity info supports "what-if" analysis.
- Inverse Problem: What LQR setup is responsible for *this* sequence (x_t, u_t)?
- NN Layers: Training by Backpropagation, a.k.a. the Chain Rule.
  For $H(\theta) = h_N \circ \cdots \circ h_3 \circ h_2 \circ h_1$,

$$D_\theta H = [D_\theta h_1]\,[D_\theta h_2]\,[D_\theta h_3] \cdots [D_\theta h_N]\,.$$

If h_k involves LQR, with $\theta_k \sim (F, f, C, c)$, the operator $D_\theta h_k$ is complicated, but to build $D_\theta H[w]$, we only need to evaluate $D_\theta h_k$, i.e., to find

$$D_\theta h_k[w_k], \quad \text{for} \quad w_k = [D_\theta h_{k+1}] \cdots [D_\theta h_N]\, w.$$

That's what this construction provides!

Implementation and Further Reading

Try it yourself:

https://github.com/PhilipLoewen/DiscreteLQR

Read more:

- Amos and Kolter, OptNet: Differentiable Optimization as a Layer in Neural Networks, ICML 2017. arXiv 1703.00443
- Bounou, Ponce, Carpentier, Leveraging Proximal Optimization for Differentiating Optimal Control Solvers, IEEE CDC 2023.

Professional Software:

- Pytorch autograd, the standard backprop wrapper.
- JAX, Google's Python library for accelerator-oriented array computation and program transformation.
- optax, c/o Google Deepmind, a gradient processing and optimization library for JAX.

Leftovers

Sidebar/Appendix – The QP Value Function

Consider $\quad V(M,m,G,g) = \min_{x\in\mathbb{R}^p} \left\{ \frac{1}{2}x^T Mx - m^T x : \; Gx = g \right\}$.

With $\widehat{x}(M,m,G,g) = \arg\min\{\cdots\}$,

$$V(M,m,G,g) = \tfrac{1}{2}\widehat{x}^T M\widehat{x} - m^T\widehat{x} = \widetilde{\ell}(\widehat{x};M,m).$$

Here $\widetilde{\ell}(x;M,m) = \frac{1}{2}x^T Mx - m^T x$ has $\nabla_x\widetilde{\ell}(x;M,m) = Mx - m$.

From above,

1. Solve for (x,λ): $\begin{bmatrix} M & G^T \\ G & 0 \end{bmatrix}\begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} m \\ g \end{bmatrix}$. Notice $Mx - m = -G^T\lambda$.
2. Solve for (z_0,z_1): $\begin{bmatrix} M & G^T \\ G & 0 \end{bmatrix}\begin{bmatrix} z_0 \\ z_1 \end{bmatrix} = \begin{bmatrix} -G^T\lambda \\ 0 \end{bmatrix}$. Inspection: $\begin{bmatrix} z_0 \\ z_1 \end{bmatrix} = \begin{bmatrix} 0 \\ -\lambda \end{bmatrix}$.
3. Report: $\begin{aligned} \nabla_m V &= -x, & \nabla_g V &= -\lambda, \\ \nabla_M V &= \tfrac{1}{2}xx^T, & \nabla_G V &= \lambda x^T. \end{aligned}$ (Reassuring.)