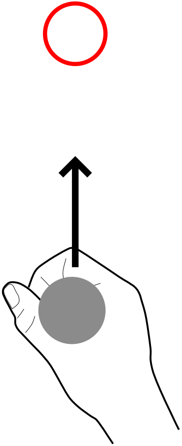
LQG Regulator and Applications to Neural Control of Movement

F. Crevecoeur

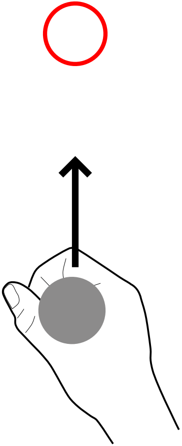
CoSMo, 2013

Control Problem: Example



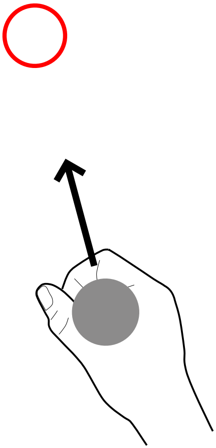
- How should you push the handle to steer it to the goal target?

Control Problem: Example



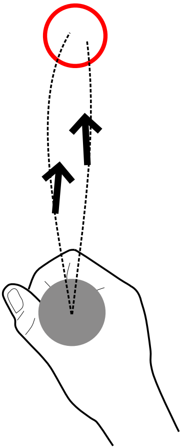
- How should you push the handle to steer it to the goal target?
- It depends on the handle dynamics (Newton's laws).

Control Problem: Example



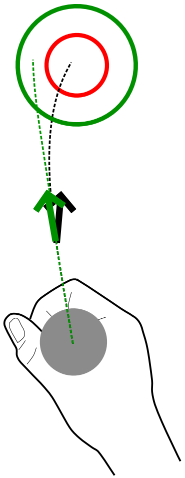
- How should you push the handle to steer it to the goal target?
- It depends on the handle dynamics (Newton's laws).
- It depends on the instantaneous *state*.

Control Problem: Example



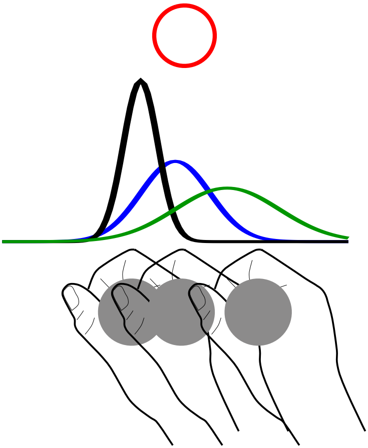
- How should you push the handle to steer it to the goal target?
- It depends on the handle dynamics (Newton's laws).
- It depends on the instantaneous *state*.

Control Problem: Example



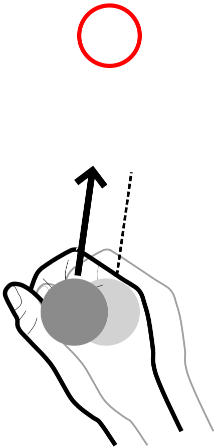
- How should you push the handle to steer it to the goal target?
- It depends on the handle dynamics (Newton's laws).
- It depends on the instantaneous *state*.
- It depends on the task, i.e. the *cost-function*.

Estimation Problem: Example



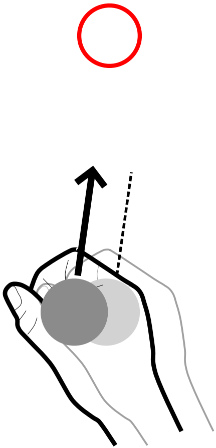
- Where is the hand?

Estimation Problem: Example



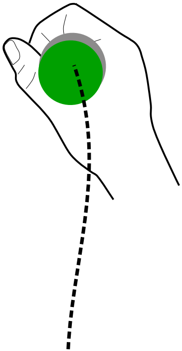
- Where is the hand?
- Combine internal priors with sensory feedback (Bayesian inference).

Estimation Problem: Example



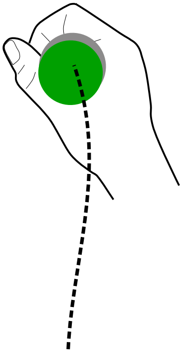
- Where is the hand?
- Combine internal priors with sensory feedback (Bayesian inference).
- Use sensory feedback to update the estimate.

Control and Estimation



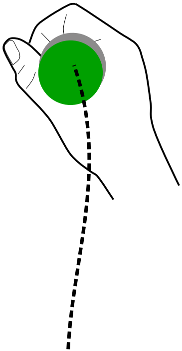
- Estimate the hand position

Control and Estimation



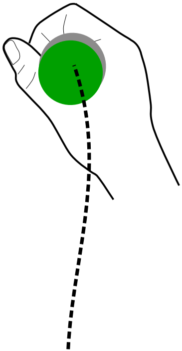
- Estimate the hand position
- Apply a force vector accordingly

Control and Estimation



- Estimate the hand position
- Apply a force vector accordingly
- Update the estimation and control processes

Control and Estimation



- Estimate the hand position
- Apply a force vector accordingly
- Update the estimation and control processes

That's it !

LQG Control Framework

- **Linear**: Linear dynamics in state and control variables,
- **Quadratic**: Quadratic cost-function in state and control variables,
- **Gaussian**: Assume that the noise variables are normally distributed ($X \sim N(\mu, \sigma^2)$).

Definitions

Control System:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + \xi_k, \\ y_k &= Hx_k + \omega_k, \end{aligned}$$

$x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$ and $y \in \mathbb{R}^p$. The initial state is given (x_1).

Cost Function:

$$\begin{aligned} J_k(x_k, u_k) &= x_k^T Q_k x_k + u_k^T R u_k, & Q_k \geq 0, \quad k = 1, 2, ...N, \\ J_N(x_N) &= x_N^T Q_N x_N. & R > 0. \end{aligned}$$

Noise Parameters:

$$\xi_k \sim N(0, \Omega_\xi), \quad \omega_k \sim N(0, \Omega_\omega).$$

Part I: Control

Optimal Control Problem:

Find a control sequence, $u_1, u_2, \dots u_{N-1}$, which minimizes:

$$J = E\left[J_N + \sum_{k=1}^{N-1} J_k(x_k, u_k)\right],$$

where $E(.)$ denotes the expected value of the arguments.

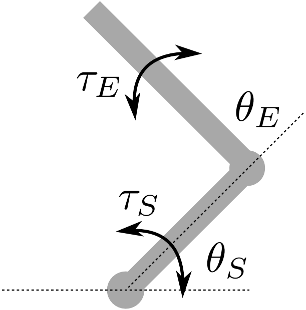
How do we solve it?

We assume for now that the controller knows the exact state of the system x_k.
It can be shown that the *cost-to-go* (i.e. the expected cost of the remaining trajectory) satisfies the Bellman equation:

$$
\begin{aligned}
v_k(x_k, u_k) &:= \min_{u_k, u_{k+1}, ..} E\left[J_N + \sum_{l=k}^{N-1} J_l(x_l, u_l)\right], \\
v_k(x_k, u_k) &= \min_{u_k}\left[J_k(x_k, u_k) + E\left(v_{k+1} | x_k, u_k\right)\right].
\end{aligned}
$$

The terminal condition is: $v_N = x_N^T Q_N x_N$. Bellman equation can be solved with dynamic programming.

Curse of Dimensionality

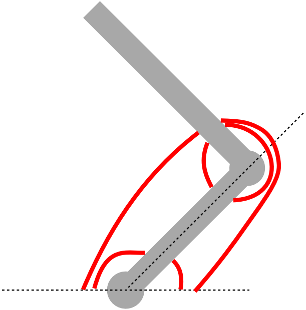


Dynamic programming is practically useless:

- State: $\text{Angles}_{S,E}$, $\text{Velocity}_{S,E}$, $\text{Torque}_{S,E}$, $\text{Torque Derivative}_{S,E}$
- $\text{Control Variables}_{S,E}$.

Considering 100 discretization points (coarse) and 50 time steps: $\sim$ h.

Curse of Dimensionality (2)



Dynamic programming is practically useless:

- State: $\text{Angles}_{S,E}$, $\text{Velocity}_{S,E}$, $\text{Torque}_{S,E}$, Torque $\text{Derivative}_{S,E}$
- 6 Control Variables.

Considering 100 discretization points (coarse) and 50 time steps: $\sim$ yrs.

Solution of the Optimal Control Problem

Theorem: Fully Observable Case. Under the optimal control policy, the cost-to-go is given by

$$v_k(x_k, u_k) = x_k^T S_k x_k + s_k,$$

with S_k are positive semidefinite and and s_k are non-negative.

Solution of the Optimal Control Problem (2)

Proof (Induction).

- The claim is true when $k = N$ with $S_N = Q_N$ and $s_N = 0$.
- Let $1 \leq k < N$. We must solve:

$$v_k = \min_{u_k} \left[x_k^T Q_k x_k + u_k^T R u_k + E\left(v_{k+1} | x_k, u_k\right) \right].$$

Expanding the conditional expected value of v_{k+1} given x_k and u_k from the induction hypothesis gives:

$$\begin{aligned} v_k &= \min_{u_k} \Big[x_k^T \left(Q_k + A^T S_{k+1} A \right) x_k + u_k^T \left(R + B^T S_{k+1} B \right) u_k \\ &\quad + 2 x_k^T A^T S_{k+1} B u_k + \mathrm{tr}\left(S_{k+1} \Omega_\xi \right) + s_{k+1} \Big]. \end{aligned}$$

Proof (Cont'd).

The previous equation is a quadratic form in u_k, which is minimized when u_k satisfies:

$$\begin{aligned} u_k &= -\left(R + B^T S_{k+1} B\right)^{-1} B^T S_{k+1} A x_k, \\ &:= -L_k x_k. \end{aligned}$$

By plugging the expression of the optimal control variable into the expression of v_k we obtain:

$$\begin{aligned} v_k &= x_k^T \left(Q_k + A^T S_{k+1}(A - BL_k)\right) x_k + s_k, \\ &:= x_k^T S_k x_k + s_k. \end{aligned}$$

where $s_k := s_{k+1} + \mathrm{tr}(S_{k+1}\Omega_\xi) > 0$. We found the required expression for v_k, and we must verify that $S_k \geq 0$ to complete the proof. ■

Control: Practical Formulation

The optimal control policy is a linear function of the state. The optimal feedback gains are given by the following backward recursion:

$$\begin{aligned} L_k &= \left(R + B^T S_{k+1} B\right)^{-1} B^T S_{k+1} A, \\ S_k &= Q_k + A^T S_{k+1}(A - BL_k), \\ s_k &= s_{k+1} + \mathrm{tr}(S_{k+1}\Omega_\xi), \\ S_N &= Q_N, \quad s_N = 0. \end{aligned}$$

The closed loop control system is described by:

$$x_{k+1} = (A - BL_k)x_k + \xi_k.$$

Comments

- The total expected cost under the optimal control policy is $v_1 = x_1^T S_1 x_1 + s_1$.

- The linear mapping of state into motor commands was not assumed a priori, it follows from linear dynamics and quadratic costs.

- Consider a one time step problem. $S_2 = Q_2$ and $u_1 = MQ_2Ax_1$ with M adequately defined. From assumptions, $Q_2 \geq 0$. Assume there exists $1 \leq j \leq n$ such that $\lambda_j(Q_2) = 0$ (dimension of the null space $\text{Ker}(Q_2)$ is ≥ 1). If $Ax_1 \in \text{Ker}(Q_2)$, then $u_1 = 0$. In other words, **if the system dynamics pushes the state in the null space of the constraints, the optimal control strategy is "DON'T DO ANYTHING".**

Signal Dependent Noise

The variability of neural signal increases with the intensity of the signal. This can be modelled by multiplicative noise:

$$x_{k+1} = Ax_k + Bu_k + \xi_k + \sum_{i=1}^{n_c} \varepsilon_i C_i u_k,$$

with C_i scaling factors and $\varepsilon_i \sim N(0,1)$. In this case, the quadratic expression in u_k contains an additional term that changes the optimal feedback gains as follows:

$$L_k = \left(R + B^T S_{k+1} B + \sum_{i=1}^{n_c} C_i^T S_{k+1} C_i \right)^{-1} B^T S_{k+1} A.$$

Signal dependent noise captures some aspects of the speed-accuracy trade-off.

Part II: Estimation

Bayes Theorem

Theorem (Bayes). Let A and B be two events, the conditional probability of B given A is:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}.$$

This can be applied to time varying stochastic processes. Let x_t be a process and y_t be a measurement of x_t. Assuming independent noise, the posterior distribution of x_t is given by:

$$f(x_t|y_t) = \frac{f(x_t|y_{t-1}, y_{t-2}, \dots)f(y_t|x_t)}{f(y_t|y_{t-1}, y_{t-2} \dots)},$$

where the prior distribution is given by:

$$f(x_t|y_{t-1}, y_{t-2}, \dots) = \int_{x_{t-1}} f(x_t|x_{t-1})f(x_{t-1}|y_{t-1}, \dots)dx_{t-1}$$

Kalman Filtering

Theorem (Kalman Filtering). Assume that (i) $x_0 \sim N(\mu_0, \Sigma_0)$, (ii) x_t and y_t satisfy:

$$\begin{aligned} x_{k+1} &= Ax_k + \xi_k \qquad \xi_k \sim N(0, \Omega_\xi) \\ y_k &= Hx_k + \omega_k \qquad \omega_k \sim N(0, \Omega_\omega), \end{aligned}$$

and (iii) ξ_k and ω_k are independent, then we have $x_{k+1} \sim N(\mu_{t+1}, \Sigma_{t+1})$ where

$$\begin{aligned} \Sigma_{k+1|k} &= A\Sigma_k A^T + \Omega_\xi, \\ K_{k+1} &= \Sigma_{k+1|k} H^T \left(H\Sigma_{k+1|k} H^T + \Omega_\omega \right)^{-1} \\ \mu_{k+1} &= A\mu_k + K_{k+1}(y_{k+1} - HA\mu_k) \\ \Sigma_{k+1} &= (I - K_{k+1} H)\, \Sigma_{k+1|k}. \end{aligned}$$

Alternative Approach: Predictive Case

Control System:

$$
\begin{aligned}
x_{k+1} &= Ax_k + Bu_k + \xi_k, \\
y_k &= Hx_k + \omega_k.
\end{aligned}
$$

We assume a convex combination of prior and feedback:

$$
\begin{aligned}
\hat{x}_{k+1} &= (1 - K) \times \text{prior} + K \times \text{feedback}, \\
\hat{x}_{k+1} &= A\hat{x}_k + Bu_k + K(y_k - H\hat{x}_k).
\end{aligned}
$$

The estimation error has the following dynamics:

$$
e_{k+1} = (A - K_k H)e_k + \xi_k - K_k\omega_k.
$$

Predictive Case (Cont'd).

The optimal Kalman gain minimize the estimation error:

$$\begin{aligned} K_k &= \arg\min_K \| e_{k+1} \|^2, \\ &= \arg\min_K \left[\mathrm{tr}\left(E(e_{k+1}e_{k+1}^T)\right)\right]. \end{aligned}$$

From the error dynamics, the terms of the error covariance matrix that depend on K_k give:

$$a(K_k) := \mathrm{tr}\left(-2K_k H\Sigma_k + K_k(H\Sigma_k H^T + \Omega_\omega)K_k^T\right),$$

which is minimized over K_k when:

$$\nabla a(K_k) = 0 \quad \Rightarrow K_k = A\Sigma_k H^T(H\Sigma_k H^T + \Omega_\omega)^{-1}.$$

Estimation: Practical Solution

The optimal state estimates and Kalman gains are obtained in a forward recursion (Σ_1 known):

$$
\begin{aligned}
\hat{x}_{k+1} &= A\hat{x}_k + Bu_k + K(y_k - H\hat{x}_k), \\
K_k &= A\Sigma_k H^T (H\Sigma_k H^T + \Omega_\omega)^{-1}, \\
\Sigma_{k+1} &= \Omega_\xi + (A - K_k H)\Sigma_k A^T.
\end{aligned}
$$

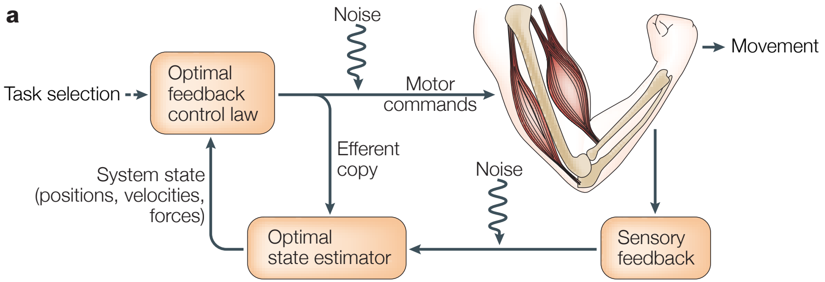
Remarks

- The control and estimation problems were solved independently and the induction proof is still valid with the state estimate (**verify it !**). This property is known as the **separation principle**. The full control systems is:

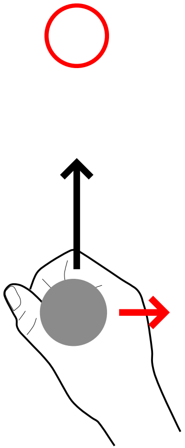
$$\begin{bmatrix} x_{k+1} \\ e_{k+1} \end{bmatrix} = \begin{bmatrix} A - BL_k & BL_k \\ 0 & A - K_k H \end{bmatrix} \begin{bmatrix} x_k \\ e_k \end{bmatrix} + \begin{bmatrix} \xi_k \\ \xi_k - K_k \omega_k \end{bmatrix}.$$

Part III: Applications

Model & Neuroscience

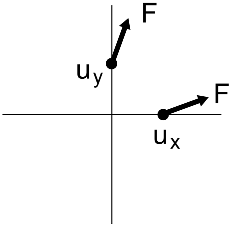


Translation of a Point Mass



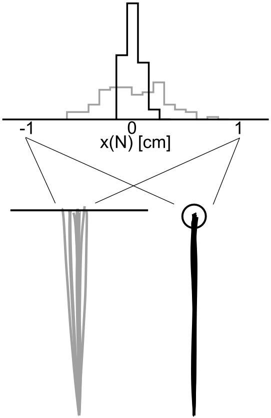
Differential Equation:

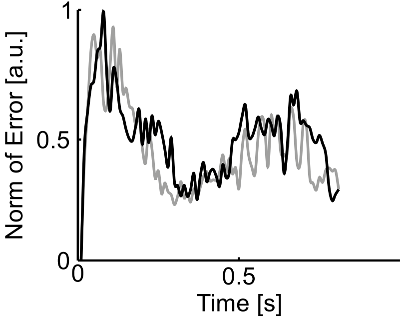
$$
\begin{aligned}
\ddot{x} &= -k_v \dot{x} + F_x \\
\ddot{y} &= -k_v \dot{y} + F_y \\
\tau \dot{F}_x &= u_x + {\color{red}\lambda_y} u_y - F_x \\
\tau \dot{F}_y &= u_y + {\color{red}\lambda_y} u_x - F_y
\end{aligned}
$$



Minimum Intervention Principle

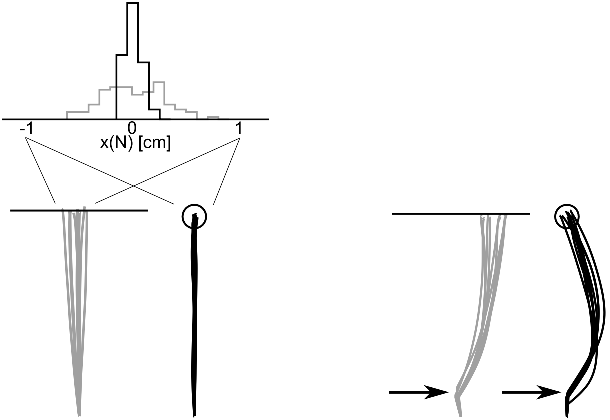
Translation of a point mass (10 cm) towards a dot (*x* and *y* constrained), or a bar (*y* constrained) in 700 ms:





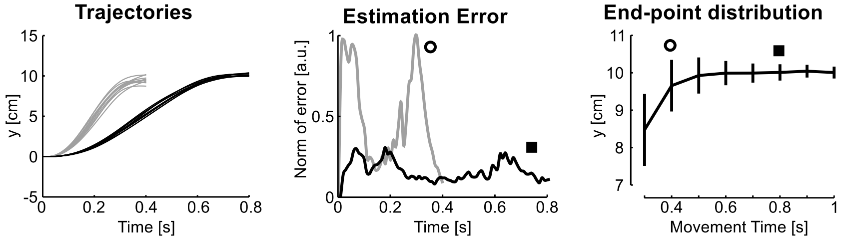
Minimum Intervention Principle (2)

The perturbation along the unconstrained dimension is left uncorrected:

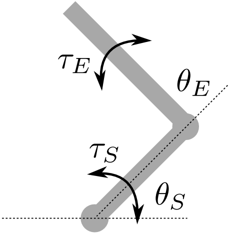


Speed Accuracy Trade-Off

Estimation error co-varies with control signals as a consequence of signal dependent noise, generating wider end-point distributions:



Approximation of Multi-Joint Dynamics



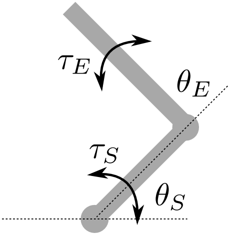
Equations of Motion:

$$\begin{aligned}\ddot{q} &= M(q)^{-1}(\tau - C(q,\dot{q}) - B\dot{q}) \\ \dot{x} &= f(x) + B(x)\tau \\ & q = [\theta_S \;\; \theta_E]^T \\ & x = [\theta_S \;\; \theta_E \;\; \dot{\theta}_S \;\; \dot{\theta}_E]^T\end{aligned}$$

A linear approximation around x_0 is given by:

$$\begin{aligned}\delta\dot{x} &= A(x_0)\delta x + B(x_0)\tau \\ & A(x_0) = \left[\frac{\partial f}{\partial x}\right]_{x_0}\end{aligned}$$

Approximation of Multi-Joint Dynamics



Equations of Motion:

$$\begin{aligned} \ddot{q} &= M(q)^{-1}(\tau - C(q,\dot{q}) - B\dot{q}) \\ \dot{x} &= f(x) + B(x)\tau \\ & q = [\theta_S \;\; \theta_E]^T \\ & x = [\theta_S \;\; \theta_E \;\; \dot{\theta}_S \;\; \dot{\theta}_E]^T \end{aligned}$$

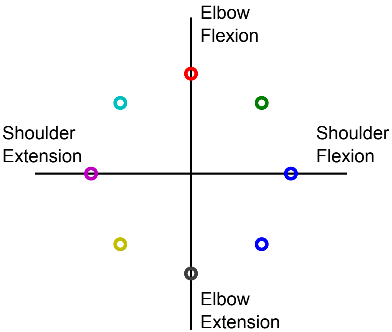
A linear approximation around x_0 is given by:

$$\begin{aligned} \delta\dot{x} &= A(x_0)\delta x + B(x_0)\tau \\ & A(x_0) = \left[\frac{\partial f}{\partial x}\right]_{x_0} \end{aligned}$$

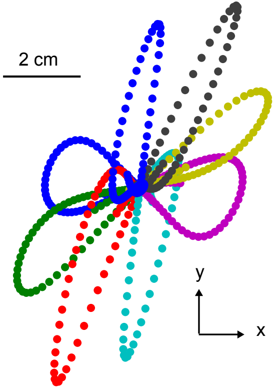
Knowledge of interaction torques (locally).

Multi-Joint Perturbations

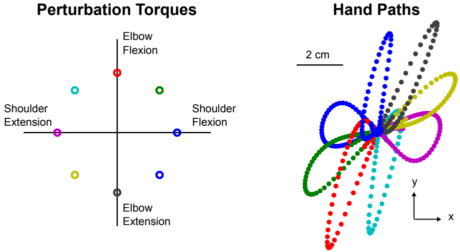
Perturbation Torques



Hand Paths



Multi-Joint Perturbations



Features:

- Internal models of dynamics (locally, and throughout the workspace)
- Noise (sensory, motor, prediction, additive and multiplicative)
- Feedback delays (system augmentation)
- Multi-sensory integration
- Task-dependent control policy

Summary

- Online control of movement can be decomposed in two process: estimation and control

Summary

- Online control of movement can be decomposed in two process: estimation and control
- In the particular case of LQG, control and estimation are independent

Summary

- Online control of movement can be decomposed in two process: estimation and control
- In the particular case of LQG, control and estimation are independent
- Optimal state estimation is obtained from a Kalman filter: a process of Bayesian integration through time

Summary

- Online control of movement can be decomposed in two process: estimation and control
- In the particular case of LQG, control and estimation are independent
- Optimal state estimation is obtained from a Kalman filter: a process of Bayesian integration through time
- The optimal control policy is a linear mapping of the estimated state into control variables

Summary

- Online control of movement can be decomposed in two process: estimation and control
- In the particular case of LQG, control and estimation are independent
- Optimal state estimation is obtained from a Kalman filter: a process of Bayesian integration through time
- The optimal control policy is a linear mapping of the estimated state into control variables
- The model captures our ability to perform goal-directed feedback control

Conclusion and Take Home Message

Pros:

- Formal link between motor behaviour (cost-function) and biomechanics (dynamical system).

Conclusion and Take Home Message

Pros:

- Formal link between motor behaviour (cost-function) and biomechanics (dynamical system).
- Powerful tool to guide experiments and interpretations

Conclusion and Take Home Message

Pros:

- Formal link between motor behaviour (cost-function) and biomechanics (dynamical system).
- Powerful tool to guide experiments and interpretations

Cons:

- Limited by numerical techniques (high-dimensional system)

Conclusion and Take Home Message

Pros:

- Formal link between motor behaviour (cost-function) and biomechanics (dynamical system).
- Powerful tool to guide experiments and interpretations

Cons:

- Limited by numerical techniques (high-dimensional system)

Agnostic:

- Neural implementation: Direct control of muscles or synergies?

Annex: Math Reminders

Tip # 1

A matrix $A \in \mathbb{R}^{n \times n}$ is positive definite if:

$$x^T A x > 0, \quad \forall x \in \mathbb{R}^n.$$

A matrix $A \in \mathbb{R}^{n \times n}$ is positive semidefinite if:

$$x^T A x \geq 0, \quad \forall x \in \mathbb{R}^n.$$

If $A > 0$, then $\lambda_i(A) > 0$ for $i = 1, 2, ...n$ and A is invertible. For a positive semidefinite matrix A, there is a manifold M embedded in $\mathbb{R}^n$ such that $Ay = 0$ for all $y \in M$. M is called the null-space, or Kernel, of A.

Tip # 2

- A probability space is a triple $(\Omega, \mathcal{U}, P)$, Ω being a set, $\mathcal{U}$ is a collection of subsets of Ω (called a σ-algebra) and P is a measure of the elements of $\mathcal{U}$.
- $A \in \mathcal{U}$ is an event, and $P(A)$ is the probability of the event A.

$$P(A) := \int_A dP.$$

- The expected value of a random variable $X \in \Omega$ is:

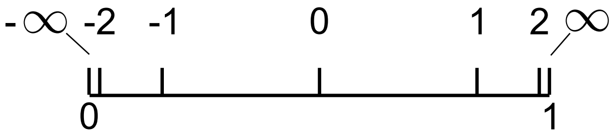
$$E(X) := \int_\Omega X dP.$$

- For a univariate Gaussian random variable, we have $\Omega = \mathbb{R}$, $\mathcal{U}$ is the collection of open and closed sets of $\mathbb{R}$ and the measure the Gaussian cumulative distribution function. The expected value is:

$$E(X) := \int_{\mathbb{R}} X f(X; \mu, \sigma^2) dX.$$

Cont'd.

- A probability density function is a measure of a space. The Normal distributions maps de real numbers into [0 1] as follows:



- The uniform distribution between 0 and 1 is given by the Lebesgue measure.

Tip # 3

The induction proof is a common tool to prove that $P(n)$ is true for all values of $n \in \mathbb{N}$.

- The initial case: show that $P(1)$ is true (easy, from assumptions).
- The induction case: assume that $P(n)$ is true with $n > 0$, show that it is also true for $P(n+1)$ (can be hard, very hard).
- If the induction case holds, than the set of $\bar{N} \subset \mathbb{N}$ such that $P(\bar{N})$ is false is between 1 and n. As n is arbitrary, we have $\bar{N} = \emptyset$.

Example: All cars are the same colour.
Proof: One car is one colour. Any $n+1$ cars is made of two overlapping subsets of n cars, with cars 1 to n and 2 to $n+1$. From the induction hypothesis that any n cars are the same colour, we have shown the the $n+1$ cars are the same colour. ■

Tip # 4

Lemma. Let x be a Gaussian random variable with mean value $\mu \in \mathbb{R}^n$ and covariance matrix $\Omega_x \in \mathbb{R}^{n \times n}$, and $S \in \mathbb{R}^{n \times n}$. Then

$$E(x^T S x) := \mu^T S \mu + \mathrm{tr}(S \Omega_x),$$

where tr(.) denotes the trace of the argument (i.e. the sum of the diagonal elements).

Tip # 5

Let $f(x)$ be a real valued function whose derivatives up to order $n+1$ exist in the neighbourhood of x_0. The n^{th} order Taylor's expansion of the $f(x)$ around x_0 is:

$$
\begin{aligned}
f(x) &\simeq \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!}(x-x_0)^k, \\
&\simeq f(x_0) + \left[\frac{df}{dx}\right]_{x_0}(x-x_0) + \frac{1}{2}\left[\frac{d^2 f}{dx^2}\right]_{x_0}(x-x_0)^2 + \ldots
\end{aligned}
$$

Euler integration is the application of Taylor's expansion to integration through time:

$$
\begin{aligned}
x(t+\delta t) &= x(t) + \dot{x}(t)\delta t + \mathcal{O}(\delta t^2), \\
&\simeq x(t) + f(x)\delta t.
\end{aligned}
$$

With linear dynamics, $f(x) = Ax$, we have:

$$
x(t+\delta t) \simeq (I_n + A\delta t)x(t).
$$

Tip # 6

Any linear n^{th} order ODE can be transformed in a n-dimensional first order ODE:

$$u^{(n)} = a_0 u + a_1 u' + \cdots + a_{n-1} u^{n-1}$$

$$\Leftrightarrow$$

$$\begin{bmatrix} u' \\ u'' \\ \vdots \\ u^{(n)} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ a_0 & a_1 & a_2 & \dots & a_{n-1} \end{bmatrix} \begin{bmatrix} u \\ u' \\ \vdots \\ u^{(n-1)} \end{bmatrix}$$

Tip # 7

Expressing spatial constraints as a quadratic function is done by augmenting the system with the target vector. Let x and x^* be the state variable and the target. Then we have

$$\| x - x^* \|^2 = \begin{bmatrix} x & y & x^* & y^* \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ x^* \\ y^* \end{bmatrix}.$$

Model Matrices

$$A_0 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -k_v & 0 & 1 & 0 \\ 0 & 0 & 0 & -k_v & 0 & 1 \\ 0 & 0 & 0 & 0 & \frac{-1}{\tau} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-1}{\tau} \end{bmatrix} \quad B_0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1/\tau & \lambda/\tau \\ \lambda/\tau & 1/\tau \end{bmatrix}$$

These matrices must be multiplied by δt and augmented to include the target vector:

$$A = \begin{bmatrix} \mathcal{I}_{6\times 6} + \delta t A_0 & \mathcal{O}_{6\times 6} \\ \mathcal{O}_{6\times 6} & \mathcal{I}_{6\times 6} \end{bmatrix} \quad B = \delta t \begin{bmatrix} B_0 \\ \mathcal{O}_{6\times 2} \end{bmatrix}$$