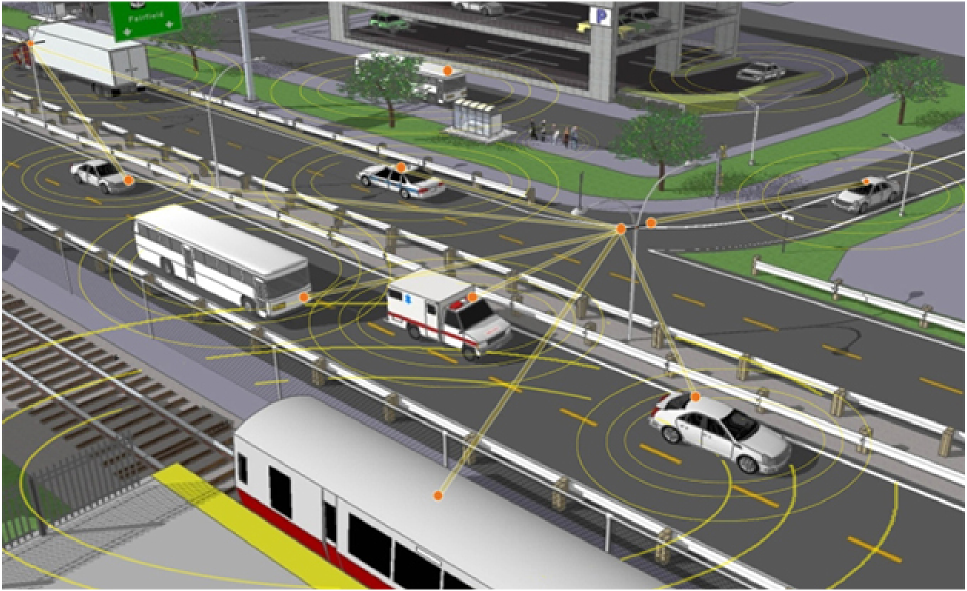
Control over Noisy Communication Media

Anatoly Khina

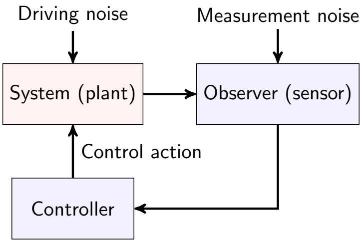
Please **DO NOT** *turn off your cell phones!*

Self-Driving Cars: Vehicle-to-Vehicle Communication

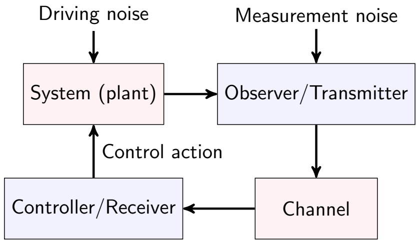


Traditional versus Networked Control

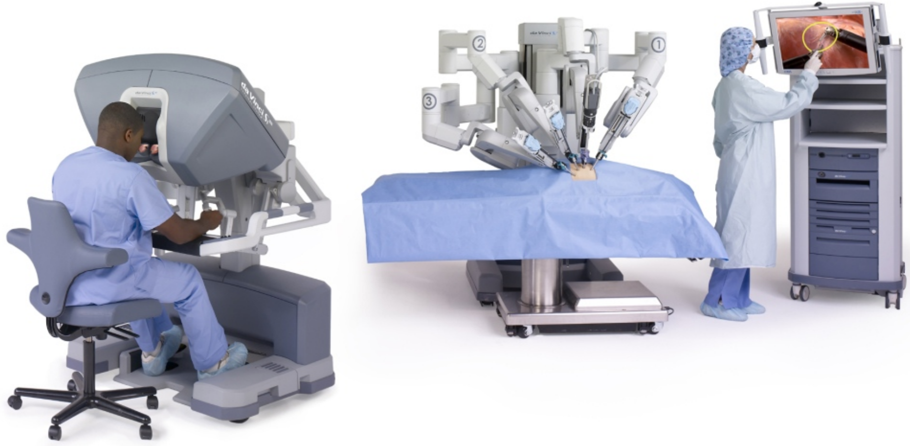
Traditional control



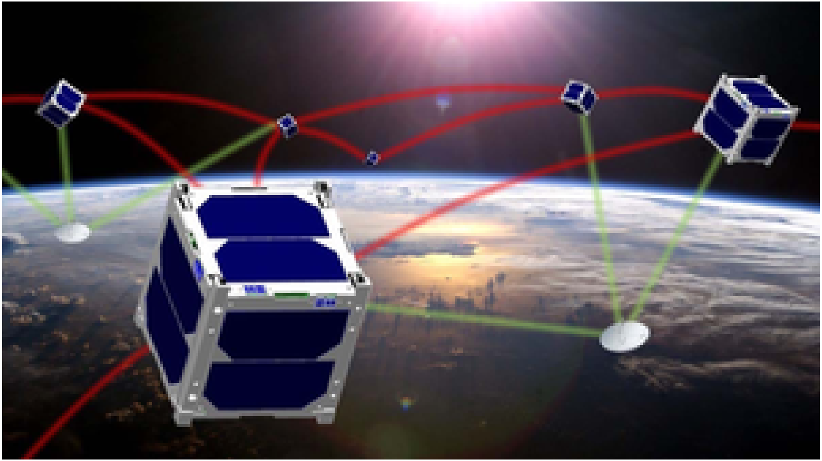
Networked control



Remote Surgery



Pico-Satellites



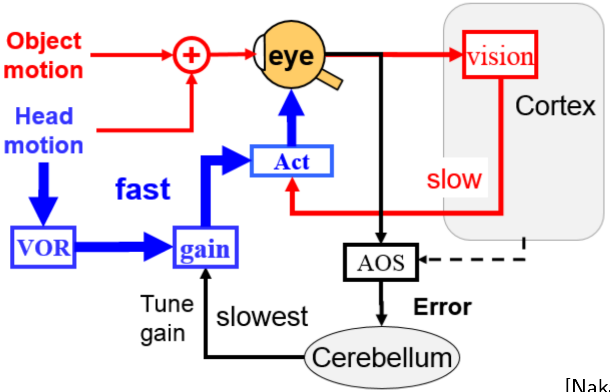
- In Israel: Genesis Consortium

Neuroscience: Resolution ⇔ Delay Tradeoff

Neuroscience: Resolution ⇔ Delay Tradeoff

Now please turn off your cell phones...

Neuroscience: Macro-level



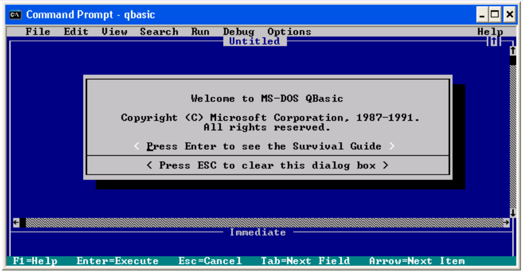
[Nakahira et al. CDC'15]

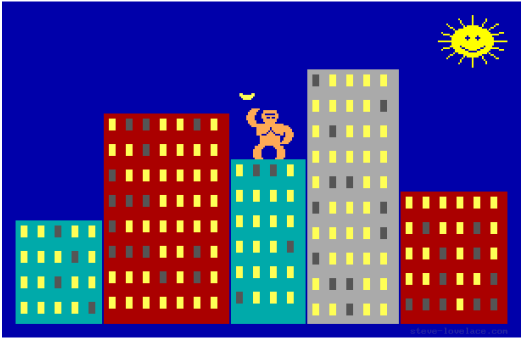
Visual system (delay $\geq$ 200ms, high res.)

VOR = Vestibulo-Ocular Reflex (delay $\approx$ 10ms, low res.)

AOS = Accessory Optical System

Back to Basics...



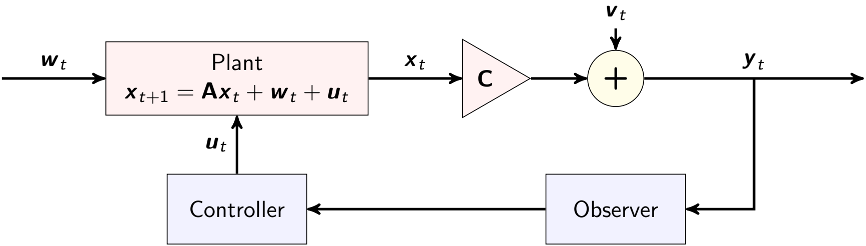


Linear Quadratic Gaussian (LQG) Control

LQG system

$$\boldsymbol{x}_{t+1} = \mathbf{A}\boldsymbol{x}_t + \mathbf{B}\boldsymbol{u}_t + \boldsymbol{w}_t, \qquad \boldsymbol{w}_t \sim \text{ i.i.d. } \mathcal{N}(0, \mathbf{W})$$

$$\boldsymbol{y}_t = \mathbf{C}\boldsymbol{x}_t + \boldsymbol{v}_t, \qquad \boldsymbol{v}_t \sim \text{ i.i.d. } \mathcal{N}(0, \mathbf{V})$$



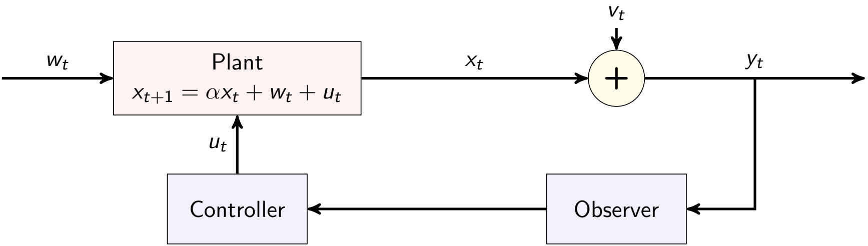
LQG cost

$$\bar{J}_T = \mathbb{E}\left[\sum_{t=1}^{T-1}\left(\boldsymbol{x}_t^T\mathbf{Q}_t\boldsymbol{x}_t + \boldsymbol{u}_t^T\mathbf{R}_t\boldsymbol{u}_t\right) + \boldsymbol{x}_T^T\mathbf{Q}_T\boldsymbol{x}_T\right]$$

Linear Quadratic Gaussian (LQG) Control

Scalar LQG system

$$x_{t+1} = \alpha x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$
$$y_t = x_t + v_t, \qquad v_t \sim \text{ i.i.d. } \mathcal{N}(0, V)$$



LQG cost

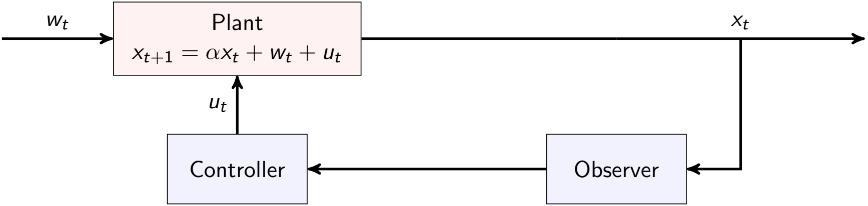
$$\bar{J}_T = \mathbb{E}\left[\sum_{t=1}^{T-1}\left(Q_t x_t^2 + R_t u_t^2\right) + Q_T x_T^2\right]$$

Linear Quadratic Gaussian (LQG) Control

Scalar linear quadratic Gaussian (LQG) system

$$x_{t+1} = \alpha x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$

$$y_t = x_t + \cancel{v_t}, \qquad \cancel{v_t \sim \text{ i.i.d. } \mathcal{N}(0, V)}$$



LQG cost

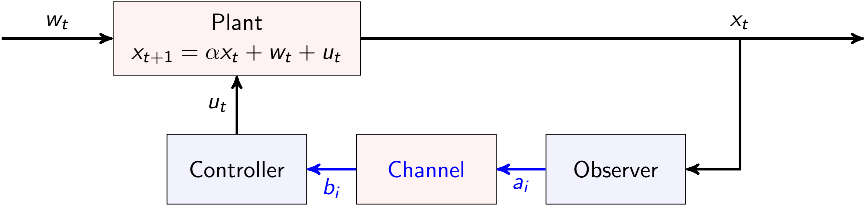
$$\bar{J}_T = \mathbb{E}\left[\sum_{t=1}^{T-1} \left(Q_t x_t^2 + R_t u_t^2\right) + Q_T x_T^2\right]$$

Linear Quadratic Gaussian (LQG) Control over Noisy Channels

Scalar linear quadratic Gaussian (LQG) system

$$x_{t+1} = \alpha x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$

$$y_t = x_t + \cancel{v_t}, \qquad \cancel{v_t} \sim \cancel{\text{i.i.d. } \mathcal{N}(0, V)}$$



LQG cost

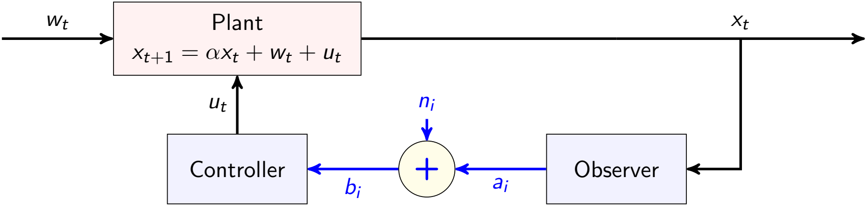
$$\bar{J}_T = \mathbb{E}\left[\sum_{t=1}^{T-1} \left(Q_t x_t^2 + R_t u_t^2\right) + Q_T x_T^2\right]$$

Linear Quadratic Gaussian (LQG) Control over Noisy Channels

Scalar linear quadratic Gaussian (LQG) system

$$x_{t+1} = \alpha x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$

$$y_t = x_t + \cancel{v_t}, \qquad \cancel{v_t} \sim \cancel{\text{i.i.d. } \mathcal{N}(0, V)}$$



LQG cost

$$\bar{J}_T = \mathbb{E}\left[\sum_{t=1}^{T-1}\left(Q_t x_t^2 + R_t u_t^2\right) + Q_T x_T^2\right]$$

Linear Quadratic Gaussian Control over Gaussian Channels

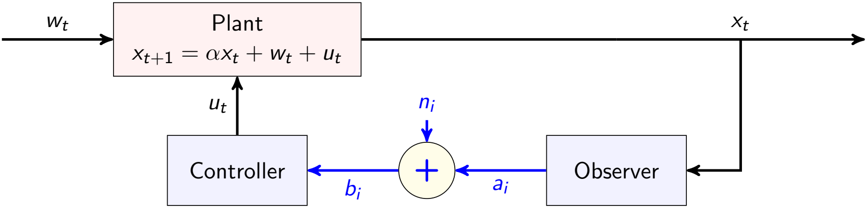
Scalar LQG system

$$x_{t+1} = \alpha x_t + u_t + w_t$$

Scalar AWGN channel

$$b_i = a_i + n_i, \qquad n_i \sim \mathcal{N}$$

Power constraint: $\mathbb{E}\left[a_i^2\right] \leq P$



Linear Quadratic Gaussian Control over Gaussian Channels

Control rate $\neq$ Communication rate!

- Assume N channel uses per one control sample

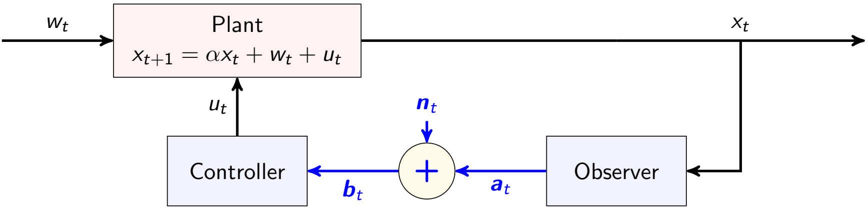
Scalar LQG system

$$x_{t+1} = \alpha x_t + u_t + w_t$$

Scalar AWGN channel

$$\boldsymbol{b}_t = \boldsymbol{a}_t\left(x^t, u^{t-1}\right) + \boldsymbol{n}_t$$

Power constraint: $\mathbb{E}\left[\boldsymbol{a}_t^2\right] \leq NP$



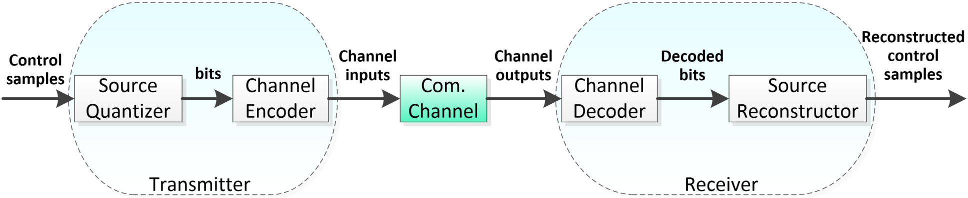
Control Sampling Rate vs. Communication Signaling Rate

- How fast the plant dynamic is $\Rightarrow$ Control sampling rate
- Bandwidth available $\Rightarrow$ Communication signaling rate
- Communication rate can be much higher in practice

How to benefit from excess signaling rate (bandwidth)?

Networked Control Approaches

Networked Control Approaches: Information-Theoretic Separation

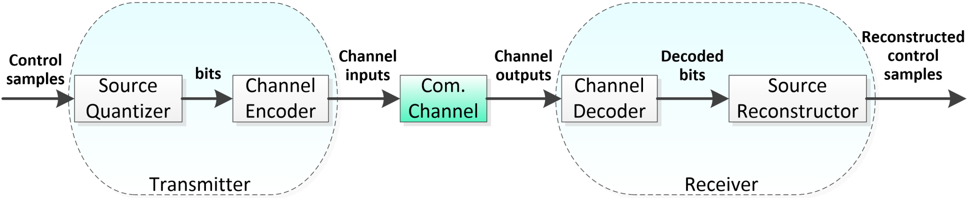


Communications is broken into two tasks:

1. Source quantization: Batch of source samples $\rightarrow$ Block of bits
2. Channel coding: Block of bits $\rightarrow$ Batch of channel uses

- Bits serve as an interface

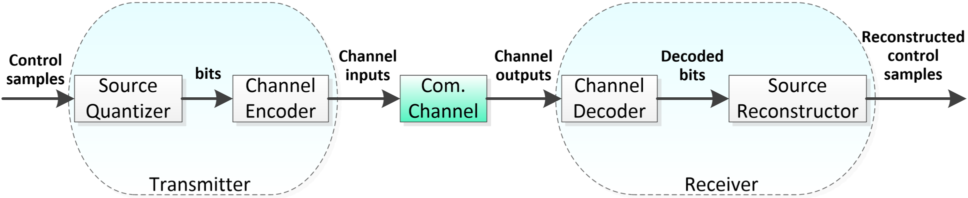
Networked Control Approaches: Information-Theoretic Separation



Advantages

- Breaks down design and analysis tasks into two simpler tasks
- Implementation: "Two different specializations"
- Breakthrough in analysis of either task $\to$ Better overall analysis
- Becomes optimal when block lengths **(=delay!)** go to infinity

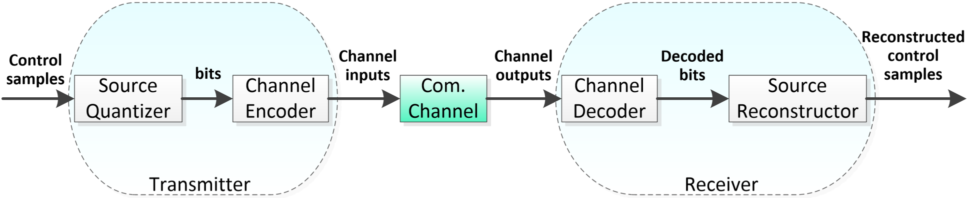
Networked Control Approaches: Information-Theoretic Separation



Shortcomings

- Requires large blocks **(delay!)** of source samples and channel uses
- **Suboptimal for control!**
- Requires codes with strong "anytime reliability" properties [Schulman IT'96][Sahai-Mitter IT'06][Sukhavasi-Hassibi AC'16]
- Problematic in practice: Convolutional code with infinite memory [Kh.-Halbawi-Hassibi ISIT'16, submitted IT'17]
- Requires proper **adaptive** quantizers: Static quantizers fail!

Networked Control Approaches: Information-Theoretic Separation



Packetizing: Extreme case

- Assumes communication rate $\gg$ control rate, very good SNR
- Problem reduces to control-oriented quantization
- Bad channel events are translated to packet drops / delays

Networked Control Approaches: Joint Source–Channel Coding (JSCC)

- What to do when control and communication rates are close?
- Can we do better than IT-separation?

Less familiar IT avenue

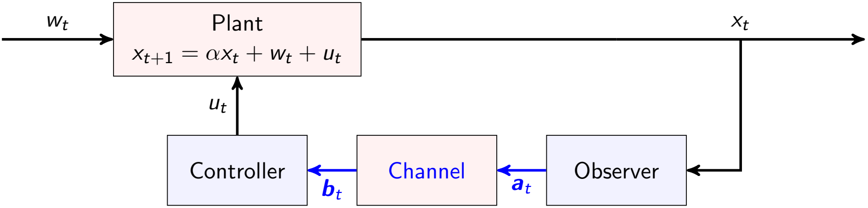
- Low-delay joint source–channel coding (JSCC)
- Analog mappings (no going through bits!)
- Control sample corresponds to source sample

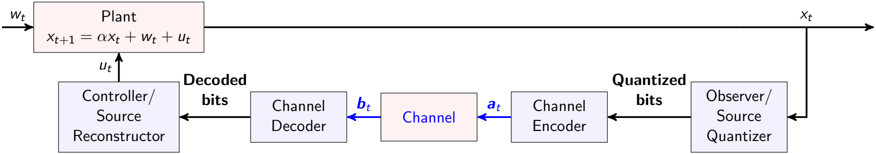
More general concept

- Use control loop as communications feedback

Source–Channel Separation

Source–Channel Separation

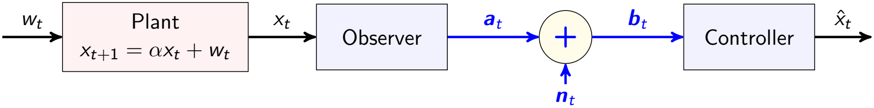




Motivating Example: Tracking a Random Walk [Sahai PhD'01]

$$x_{t+1} = \alpha x_t + w_t$$

- $|\alpha| > 1 \implies$ not stable!
- $w_t \in \{\pm 1\}$ — quantized bits representing the control state
- We wish to track x_t with bounded expected distortion
- If tracking is possible $\Rightarrow$ Stability



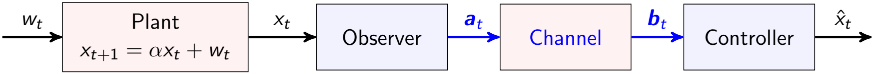
Distortion requirement

$$\mathbb{E}\left[(x_t - \hat{x}_t)^2\right] < \infty, \qquad \forall t$$

Motivating Example: Tracking a Random Walk [Sahai PhD'01]

$$x_{t+1} = \alpha x_t + w_t$$

- $|\alpha| > 1 \implies$ not stable!
- $w_t \in \{\pm 1\}$ — quantized bits representing the control state
- We wish to track x_t with bounded expected distortion
- If tracking is possible $\Rightarrow$ Stability



Distortion requirement

$$\mathbb{E}\left[(x_t - \hat{x}_t)^2\right] < \infty, \qquad \forall t$$

Motivating Example: Tracking a Random Walk [Sahai PhD'01]

- $\hat{w}_{t-d|t}$ – Estimate of w_{t-d} at time t
- Probability of first error event at time $t-d$:
  $P_e(t,d) \triangleq \Pr\left(w_{t-d} \neq \hat{w}_{t-d|t}, \forall \delta > d, w_{t-\delta} = \hat{w}_{t-\delta|t}\right)$

$$\mathbb{E}\left[\left(x_t - \hat{x}_{t|t}\right)^2\right] \propto \sum_{d=1}^{t} P_e(t,d)\alpha^{2d} = \sum_{d=1}^{t} P_e(t,d) 2^{2\log\alpha \cdot d} < \infty$$

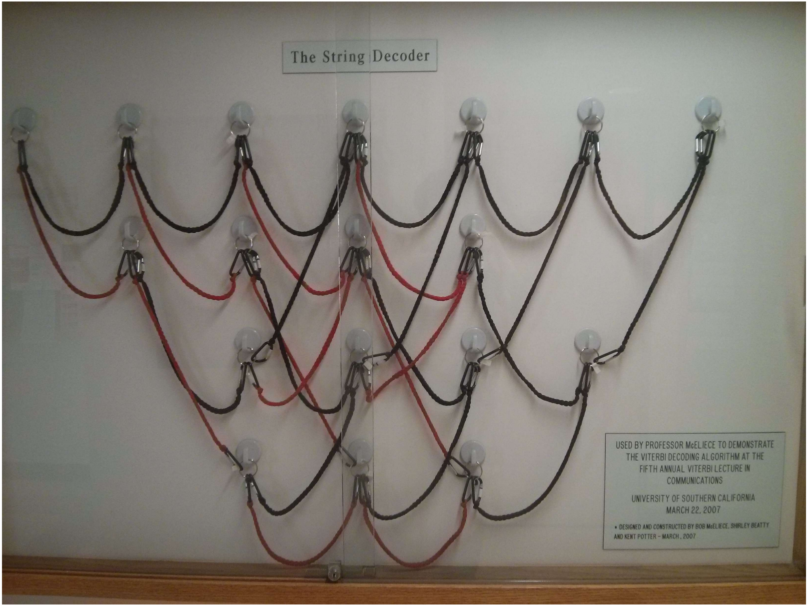
Error probability profile: Anytime-reliable code

$$P_e(t,d) < A 2^{-(2\log\alpha + \epsilon)d}, \qquad \forall t, \forall d$$

Higher-order moments

Higher exponent $\Longrightarrow$ Cannot stabilize all moments!

Tree Codes [Schulman IT'96]



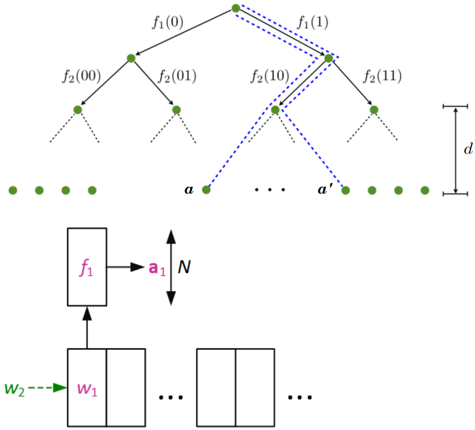
Adaptation to Control: Anytime-Reliable Codes [Sahai-Mitter IT'06]

Error probability profile

$$P_e(t,d) < A2^{-(2\log\alpha+\epsilon)d}\,, \qquad \forall t, \forall d$$

How to generate such a code?

$$\begin{aligned} a_1 &= f_1(w_1) \\ a_2 &= f_2(w_1, w_2) \\ &\vdots \\ a_t &= f_t(w_1, w_2, \ldots, w_t) \\ &\vdots \end{aligned}$$



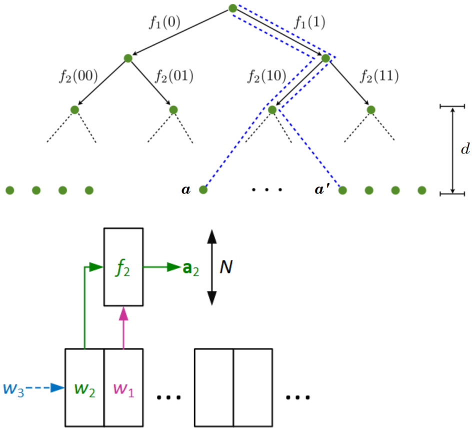
Adaptation to Control: Anytime-Reliable Codes [Sahai-Mitter IT'06]

Error probability profile

$$P_e(t,d) < A2^{-(2\log\alpha+\epsilon)d}\,, \qquad \forall t, \forall d$$

How to generate such a code?

$$\begin{aligned} \boldsymbol{a}_1 &= f_1(w_1) \\ \boldsymbol{a}_2 &= f_2(w_1, w_2) \\ &\vdots \\ \boldsymbol{a}_t &= f_t(w_1, w_2, \ldots, w_t) \\ &\vdots \end{aligned}$$



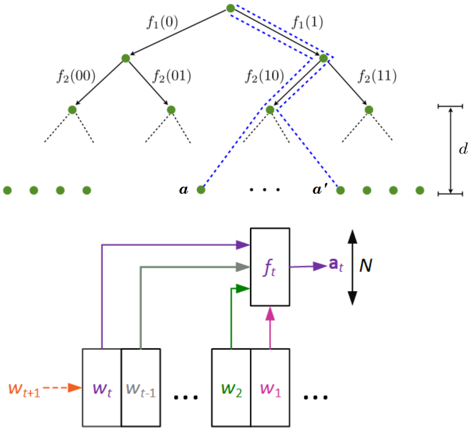
Adaptation to Control: Anytime-Reliable Codes [Sahai-Mitter IT'06]

Error probability profile

$$P_e(t,d) < A2^{-(2\log\alpha+\epsilon)d}, \qquad \forall t, \forall d$$

How to generate such a code?

$$\begin{aligned} \boldsymbol{a}_1 &= f_1(w_1) \\ \boldsymbol{a}_2 &= f_2(w_1, w_2) \\ &\vdots \\ \boldsymbol{a}_t &= f_t(w_1, w_2, \ldots, w_t) \\ &\vdots \end{aligned}$$



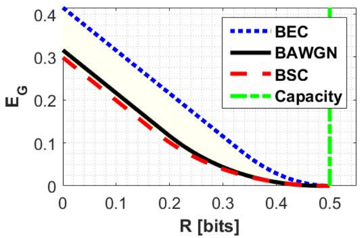
Anytime-Reliable Codes as Convolutional Codes

Random **time-varying** convolutional-code ensemble [Viterbi, Yudkin, Zigangirov, Schulman–Feder, ...]

- Most results assume infinite stream ($\gg$ delay-line length)
- We wish to recover a bit using subsequent Nd output symbols
- The random time-varying CC ensemble achieves:

$$\mathbb{E}\left[P_e(t,d)\right] \leq 2^{-E_G(R)Nd}$$

- $E_G(R) > 0$ for $R < C$ – Gallager's error exponent



Anytime-Reliable Codes as Convolutional Codes

Good ensemble performance ⇒ Good specific code performance?

- $\mathbb{E}\left[P_e(t,d)\right] \leq 2^{-E_G(R)Nd} \overset{?}{\Rightarrow} P_e(t,d) \leq A2^{-E_G(R)Nd}$
- Yes, with high probability, for **specific** t and d
- Anytime reliable-code?
- Needs to hold $\forall d$ and $\forall t$!
- Such a code exists [Schulman IT'96], but **not w.h.p.** ☹
  (Proof requires min-distance $\propto$ delay)

- LDPC-based constructions: [Grosjaen et al. IT'14] [Noor-A-Rahim et al. COM'15][Zhang et al. IT'16]
- Explicit constructions: [Gelles-Moitra-Sahai, FOCS'11, IT'14] [Moore-Schulman ITS'14][Pudlák LinAlg&Apps'16]

Anytime-Reliable Codes as Convolutional Codes

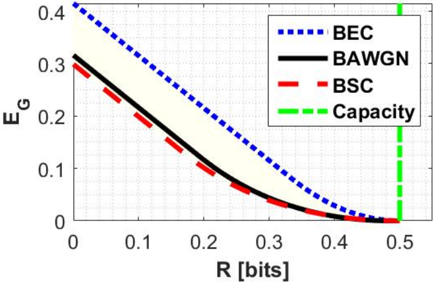
Linear time-invariant codes [Sukhavasi-Hassibi AC'16]

- Time invariance ⇒ No dependence on t: $P_e(t,d) \equiv P_e(d)$
- Proof simply follows by the union bound
- [Kh.-Halbawi-Hassibi ISIT'16, submitted IT'17]:
  Easily proved by viewing as CC + [S~~c~~hulman-Feder IT'00]
- Better results for lower rates using linear codes [Barg-Forney IT'02]

Universality [Kh.-Halbawi-Hassibi, submitted IT'17]

- High probability proof ⇒ Universality result w.r.t. channel
- Similar to the universal LDPC code construction of [Kh.-Yona-Erez ISIT'15]

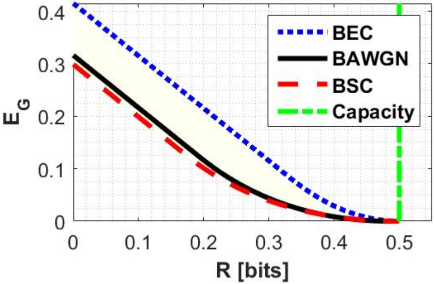
Anytime-Reliable Codes as Convolutional Codes



Universality [Kh.-Halbawi-Hassibi, submitted IT'17]

- High probability proof ⇒ Universality result w.r.t. channel
- Similar to the universal LDPC code construction of [Kh.-Yona-Erez ISIT'15]

Anytime-Reliable Codes as Convolutional Codes



Universality [Kh.-Halbawi-Hassibi, submitted IT'17]

- High probability proof ⇒ Universality result w.r.t. channel
- Similar to the universal LDPC code construction of [Kh.-Yona-Erez ISIT'15]

What about decoding?

Decoding of LTI Anytime-Reliable Codes

- All results assumed maximum-likelihood (ML) decoding
- ML complexity rises exponentially with t

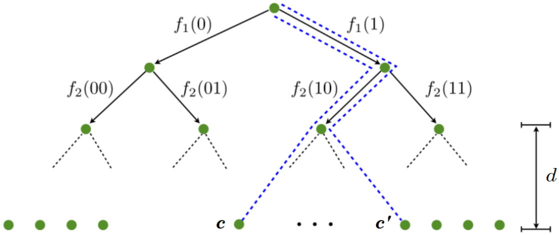
Binary Erasure Channel (BEC)

- For LTI codes: ML $=$ Solving linear equations

- What about other channels?

Sequential Decoding

- Before Viterbi algo.: Sequential decoding *de facto* standard
- Sequential decoding = class of algorithms
- Introduced originally in [Wozencraft '57] for **tree codes**
- Common to all: Explore only subset of (likely) codewords
- Most prominent variants: Stack and Fano's algorithms
- Proposed for general tree ensembles in [Schulman IT'96][Sahai-Palaiyanur Allerton'05]

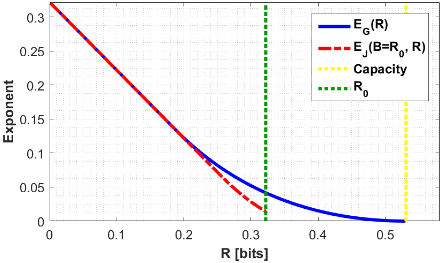


Sequential Decoding: Error Probability

Error probability of **general** conv. ensemble [Jelinek's Book '68]

$$\mathbb{E}\left[P_e(t,d)\right] \le A\, 2^{-E_J(B,R)Nd}$$

- A is finite for $B < R_0$
- $E_J(B,R) \le E_G(R)$
- $E_J(B,R) \xrightarrow{B \to R_0} E_G(R)$, for $R < R_{\text{crit}}$
- Does not guarantee a good **specific code** w.h.p.



Sequential Decoding: Error Probability

Error probability of **general** conv. ensemble [Jelinek's Book '68]

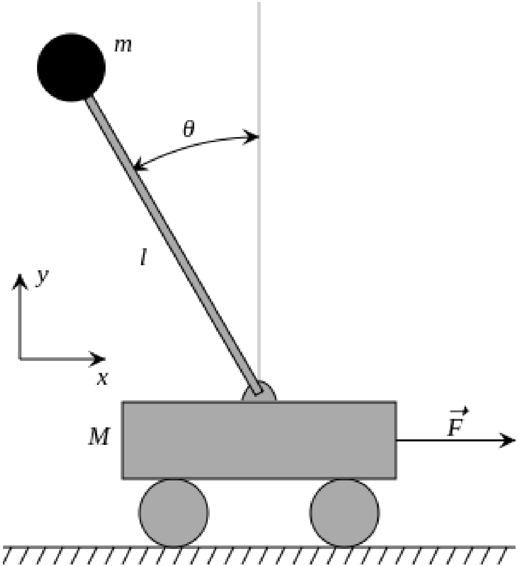
$$\mathbb{E}\left[P_e(t,d)\right] \le A\,2^{-E_J(B,R)Nd}$$

- A is finite for $B < R_0$
- $E_J(B,R) \le E_G(R)$
- $E_J(B,R) \xrightarrow{B \to R_0} E_G(R)$, for $R < R_{\text{crit}}$
- Does not guarantee a good **specific code** w.h.p.

BER of LTI tree codes [Kh.-Halbawi-Hassibi ISIT'16, submitted IT'17]

- BER guarantees extend to LTI codes
- Anytime reliable w.h.p. for a **specific code**
- Universal for channels with given capacity
- Design for the BSC

Simulation: Cart–Stick over BSC(0.01)



Simulation: Cart–Stick over BSC(0.01)

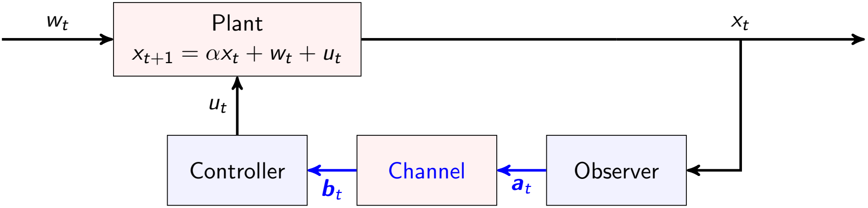
- Cart–stick system model [Franklin-Powell-Emami-Naeini Book]
- BSC(0.01)
- For this setting [Sukhavasi–Hassibi ISIT'11]: $E_{\text{min}} = 0.21$

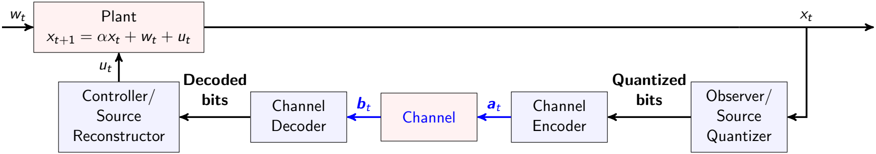
$$E = 0.54 \qquad E = 0.24 \qquad E = 0$$

Quantization

- Channel error correction ✓
- What about quantization?

Source–Channel Separation

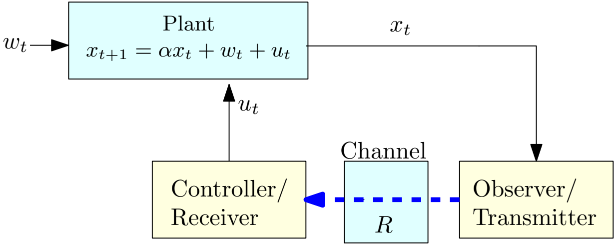




Linear Quadratic Gaussian Control over Gaussian Channels

Scalar Linear quadratic Gaussian (LQG) system

$$x_{t+1} = x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$



Noiseless finite-rate channel of rate R

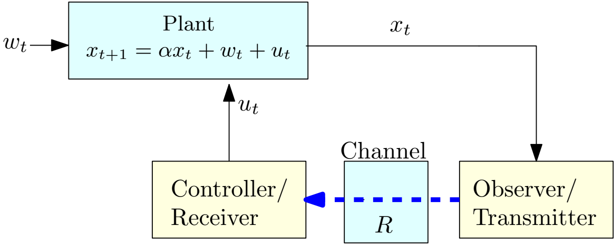
Fixed rate: Exactly R bits are available at every time step t

Variable rate: R bits are available **on average** at every t

Linear Quadratic Gaussian Control over Gaussian Channels

Scalar Linear quadratic Gaussian (LQG) system

$$x_{t+1} = x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$



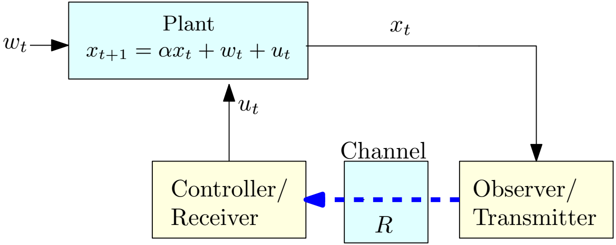
Variable-rate coding: R bits are available **on average** at every t

Elia-Mitter AC'01, Tatikonda-Sahai-Mitter AC'04, Nair-Evans SICON'04, Nair et al. ProcIEEE'07, Silva-Derpich-Østergaard AC'11, Kaspi-Merhav IT'12, Charalambous et al. AC'14, Rabi et al. SICON'16, Silva et al. AC'16, Kostina-Hassibi AC'17, Tanaka et al. AC'17, Wu-Dumitrescu ITW'17, Kh.-Kostina-Khisti-Hassibi ITW'17 & submitted TCNS'17, ...

Linear Quadratic Gaussian Control over Gaussian Channels

Scalar Linear quadratic Gaussian (LQG) system

$$x_{t+1} = x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$



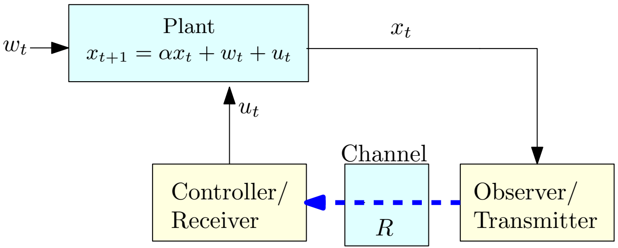
Variable-rate coding: R bits are available **on average** at every t

Elia-Mitter AC'01, Tatikonda-Sahai-Mitter AC'04, Nair-Evans SICON'04, Nair et al. ProcIEEE'07, Silva-Derpich-Østergaard AC'11, Kaspi-Merhav IT'12, Charalambous et al. AC'14, Rabi et al. SICON'16, Silva et al. AC'16, Kostina-Hassibi AC'17, Tanaka et al. AC'17, Wu-Dumitrescu ITW'17, Kh.-Kostina-Khisti-Hassibi ITW'17 & submitted TCNS'17, ...

Linear Quadratic Gaussian Control over Gaussian Channels

Scalar Linear quadratic Gaussian (LQG) system

$$x_{t+1} = x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$



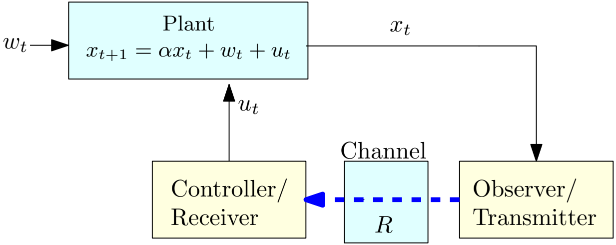
Fixed-rate coding: **exactly** R bits are available at every t

Witsenhausen BSTJ'79, Gaarder-Slepian IT'82, Walrand Varaiya IT'83, Gábor-Györfi '86, Brockett-Liberzon AC'00, Borkar-Mitter-Tatikonda SICON'01, Tatikonda-Mitter AC'04, Dumitrescu-Wu IT'09, Teneketzis IT'06, Minero-Franceschetti-Dey-Nair AC'09, Yüksel AC'10 & AC'14, Linder-Yüksel IT'14, Kh.-Nakahira-Yu-Hassibi CDC'17, ...

Linear Quadratic Gaussian Control over Gaussian Channels

Scalar Linear quadratic Gaussian (LQG) system

$$x_{t+1} = x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$



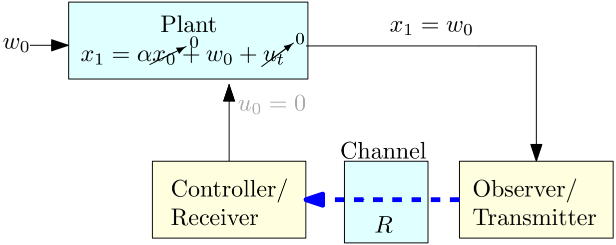
Fixed-rate coding: **exactly** R bits are available at every t

Witsenhausen BSTJ'79, Gaarder-Slepian IT'82, Walrand Varaiya IT'83, Gábor-Györfi '86, Brockett-Liberzon AC'00, Borkar-Mitter-Tatikonda SICON'01, Tatikonda-Mitter AC'04, Dumitrescu-Wu IT'09, Teneketzis IT'06, Minero-Franceschetti-Dey-Nair AC'09, Yüksel AC'10 & AC'14, Linder-Yüksel IT'14, Kh.-Nakahira-Yu-Hassibi CDC'17, ...

Linear Quadratic Gaussian Control over Gaussian Channels

Scalar Linear quadratic Gaussian (LQG) system

$$x_{t+1} = x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$



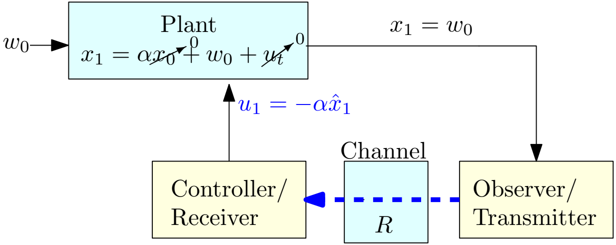
LQG cost

$$J = \mathbb{E}\left[\sum_{t=1}^{T}\left[Q_t x_t^2 + R_t u_t^2\right] + Q_T x_{T+1}^2\right]$$

Linear Quadratic Gaussian Control over Gaussian Channels

Scalar Linear quadratic Gaussian (LQG) system

$$x_{t+1} = x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$



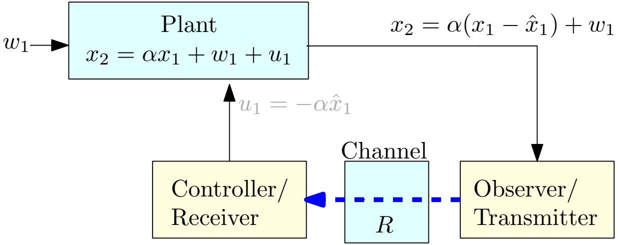
LQG cost: MMSE ($Q_t \equiv 1, R_t = 0$)

$$J = \mathbb{E}\left[\sum_{t=1}^{T+1} x_t^2\right]$$

Linear Quadratic Gaussian Control over Gaussian Channels

Scalar Linear quadratic Gaussian (LQG) system

$$x_{t+1} = x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$



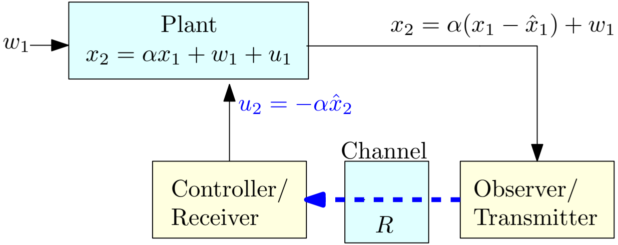
LQG cost: MMSE ($Q_t \equiv 1, R_t = 0$)

$$J = \mathbb{E}\left[\sum_{t=1}^{T+1} x_t^2\right]$$

Linear Quadratic Gaussian Control over Gaussian Channels

Scalar Linear quadratic Gaussian (LQG) system

$$x_{t+1} = x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$



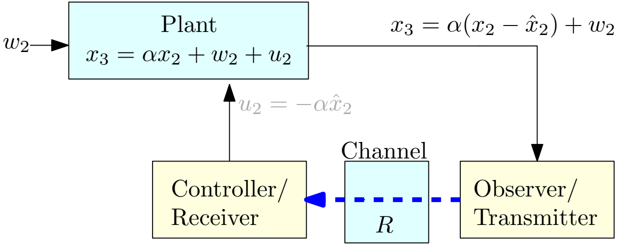
LQG cost: MMSE ($Q_t \equiv 1, R_t = 0$)

$$J = \mathbb{E}\left[\sum_{t=1}^{T+1} x_t^2\right]$$

Linear Quadratic Gaussian Control over Gaussian Channels

Scalar Linear quadratic Gaussian (LQG) system

$$x_{t+1} = x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$



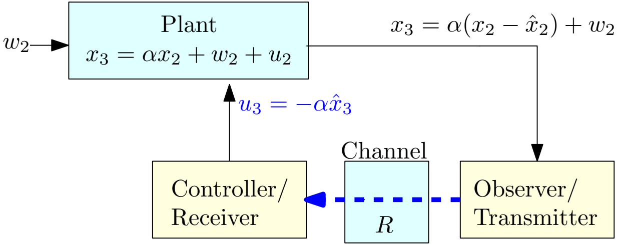
LQG cost: MMSE ($Q_t \equiv 1, R_t = 0$)

$$J = \mathbb{E}\left[\sum_{t=1}^{T+1} x_t^2\right]$$

Linear Quadratic Gaussian Control over Gaussian Channels

Scalar Linear quadratic Gaussian (LQG) system

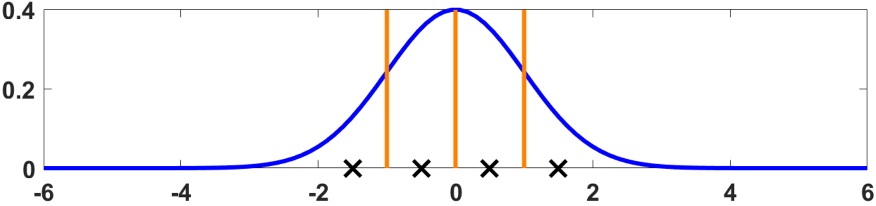
$$x_{t+1} = x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$



LQG cost: MMSE ($Q_t \equiv 1, R_t = 0$)

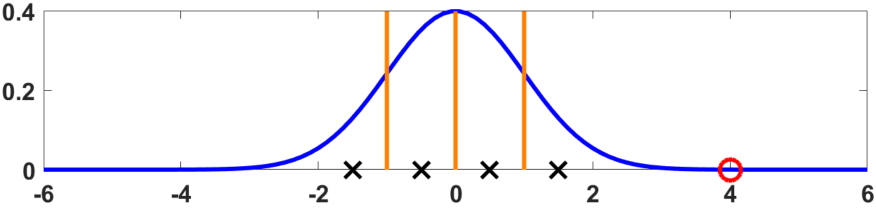
$$J = \mathbb{E}\left[\sum_{t=1}^{T+1} x_t^2\right]$$

Adaptive Fixed-Rate Quantizer



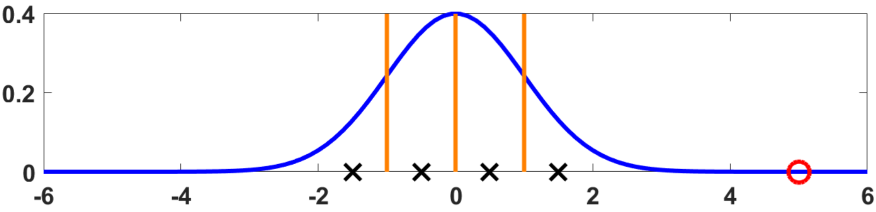
- Use an adjusted quantizer to the input p.d.f.

Adaptive Fixed-Rate Quantizer



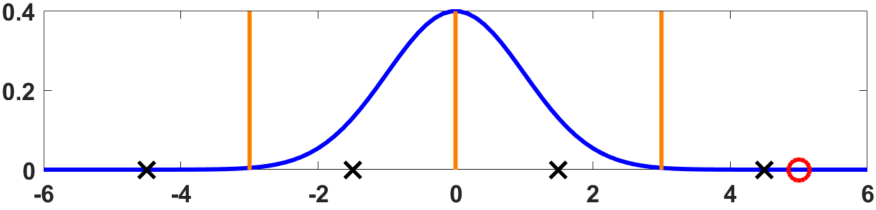
- Use an adjusted quantizer to the input p.d.f.
- At some point a (rare) event will happen
- Input value outside effective quantization interval

Adaptive Fixed-Rate Quantizer



- Use an adjusted quantizer to the input p.d.f.
- At some point a (rare) event will happen
- Input value outside effective quantization interval
- Next time step: Input will be even larger!
- **Avalanche effect**

Adaptive Fixed-Rate Quantizer



- Use an adjusted quantizer to the input p.d.f.
- At some point a (rare) event will happen
- Input value outside effective quantization interval
- Next time step: Input will be even larger!
- **Avalanche effect**
- To avoid this $\Rightarrow$ Quantizer needs to be **adaptive**

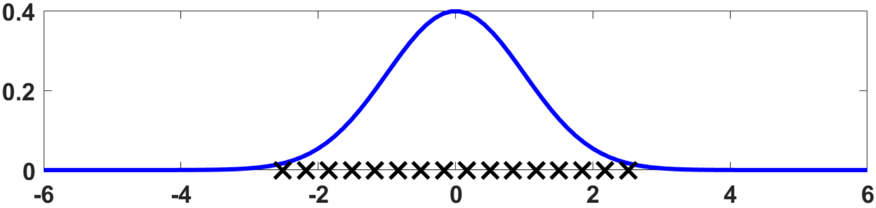
Adaptive Optimal Fixed-Rate Quantizer?

- Adaptive uniform quantizer [Yüksel AC'10]
  - Based on Jayant's adaptive quantizer [Jayant '73]
  - Similar idea in [Brockett-Liberzon AC'00]: "Zooming in/out"
- Adaptive exponential quantizer [Nair-Evans SICON'04][Minero et al. AC'09]
- Both results prove condition on stabilizability: $R > \log \alpha$
- But no cost optimality claims...
- Other notable contributions: [Borkar-Mitter '97] [Tatikonda-Sahai-Mitter AC'04] [Matveev-Savkin '04] [Tsumura-Maciejowski CDC'03], ...

How to optimize cost?

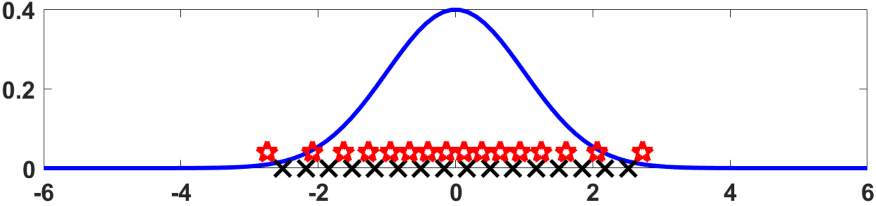
Optimal Quantizer for One Time Step

- Let $x \sim \mathcal{N}(0, 1)$
- R bits $\Rightarrow 2^R$ quantization points
- Uniform quantizer is suboptimal
- How to construct an optimal quantizer?



Optimal Quantizer for One Time Step

- Let $x \sim \mathcal{N}(0,1)$
- R bits $\Rightarrow 2^R$ quantization points
- Uniform quantizer is suboptimal
- How to construct an optimal quantizer?



- In general a hard (NP-hard) problem
- Necessary conditions by Lloyd ['57, IT'82] and Max [IT '60]
  - Also known in machine learning as "k-means" clustering

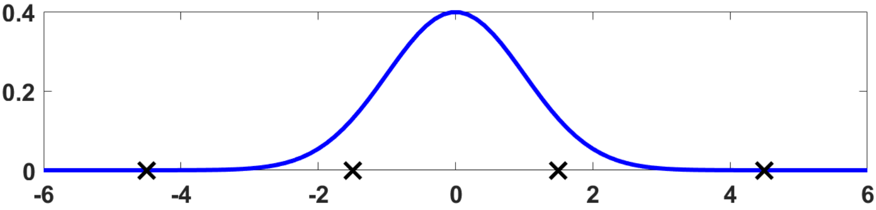
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



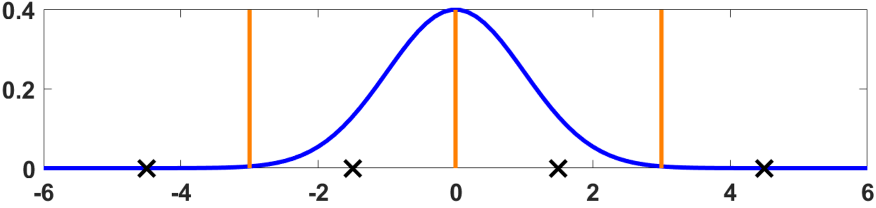
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



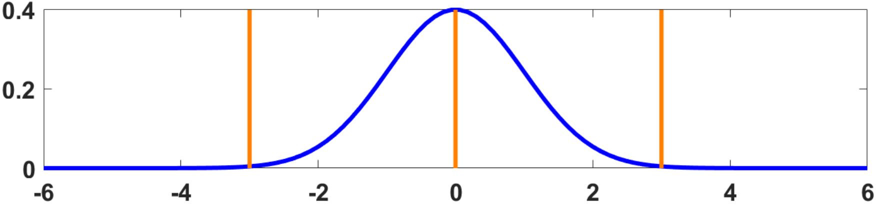
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



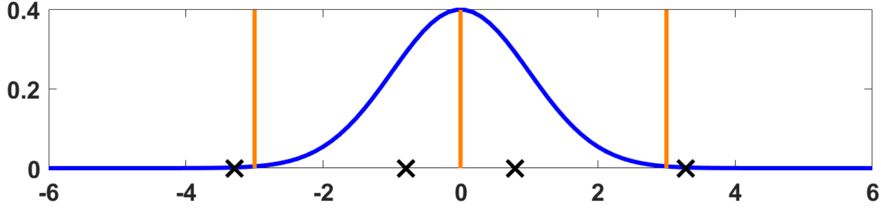
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2,\ \forall j \neq i\right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



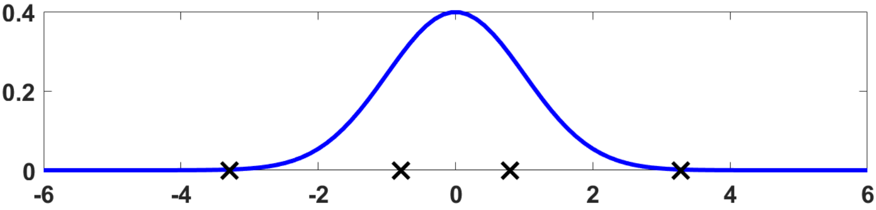
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



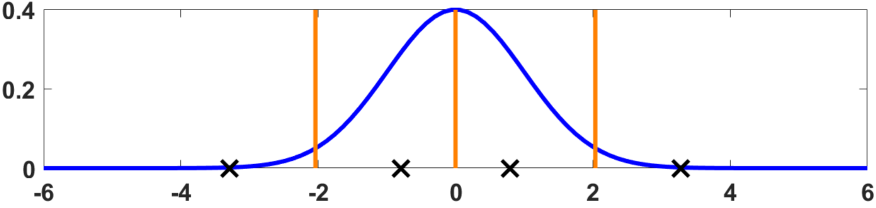
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



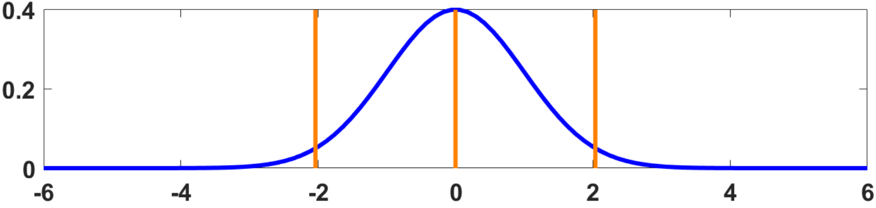
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i\right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



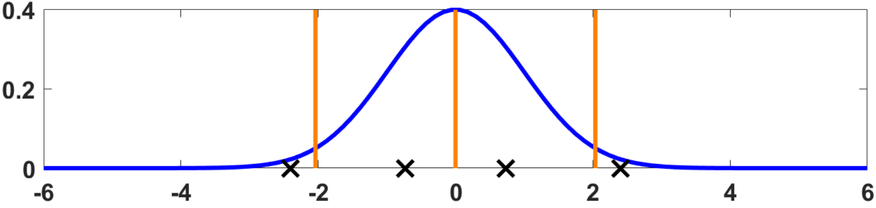
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



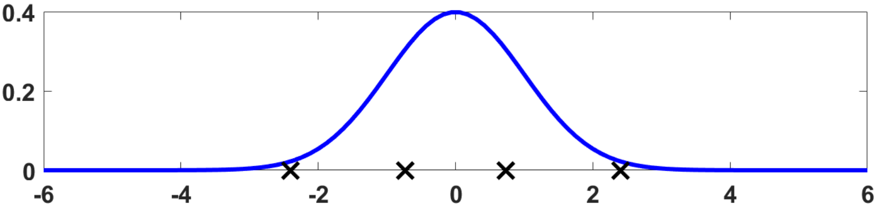
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i\right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



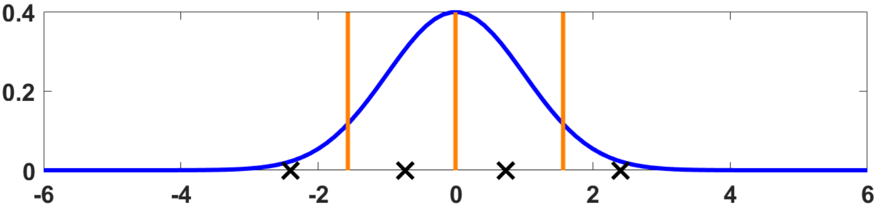
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



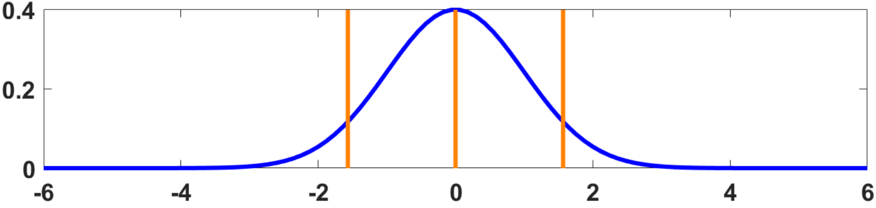
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



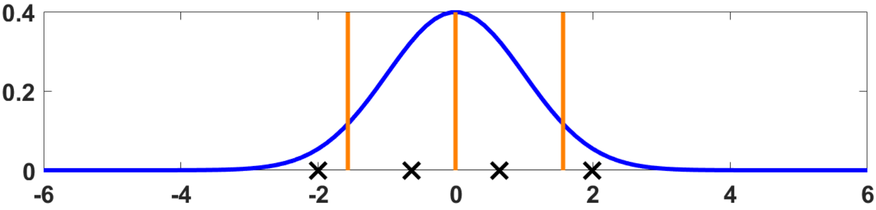
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



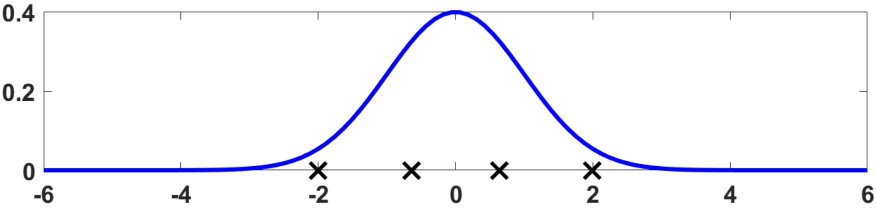
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



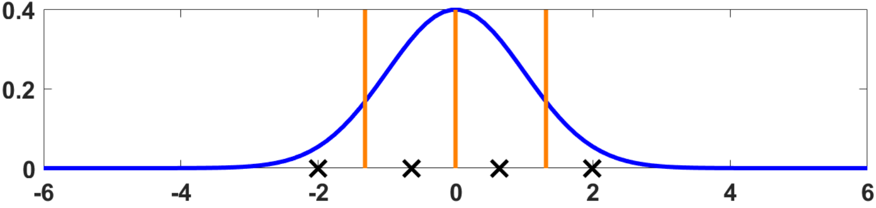
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2,\ \forall j \neq i\right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



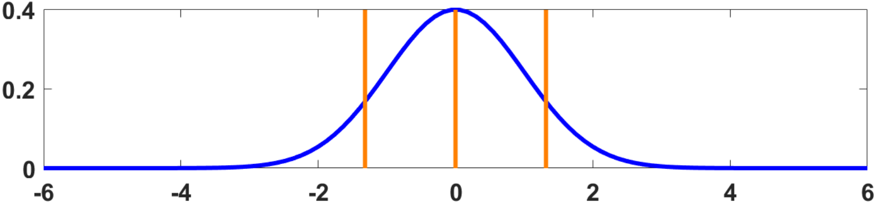
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



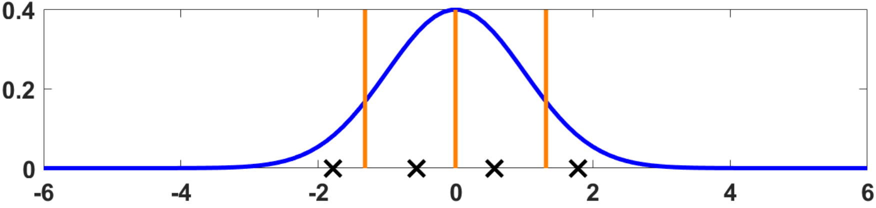
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



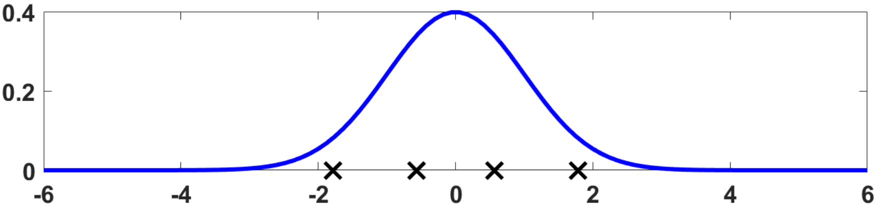
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



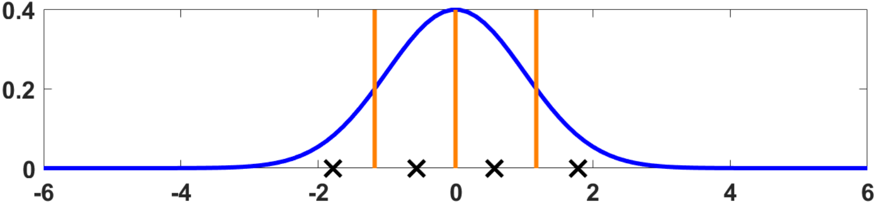
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i\right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



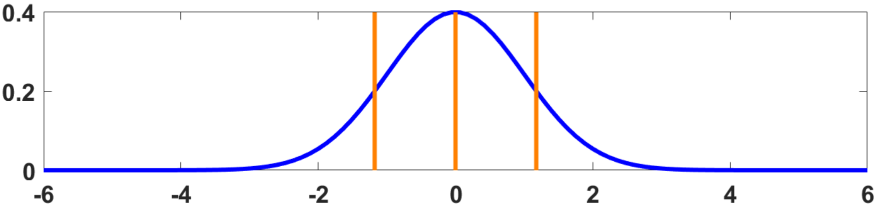
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



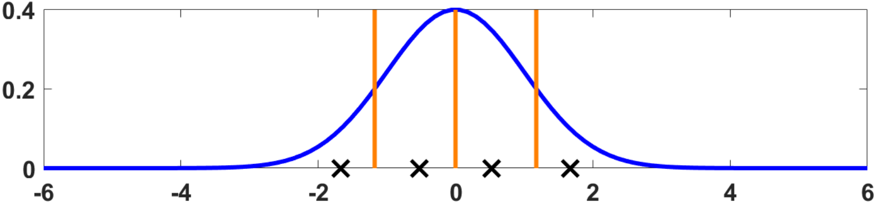
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



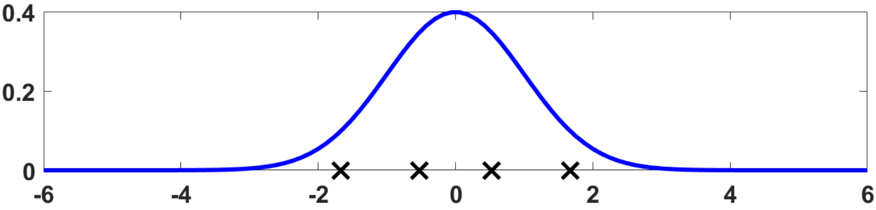
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



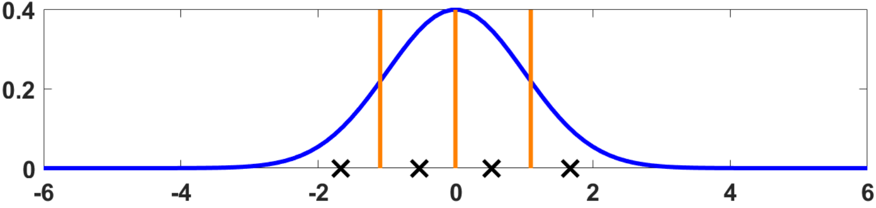
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x-\hat{x}_i)^2 < (x-\hat{x}_j)^2,\ \forall j \neq i\right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



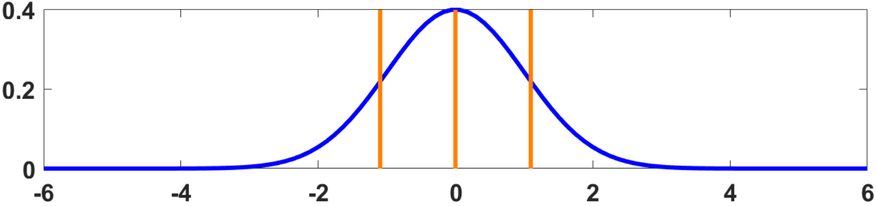
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



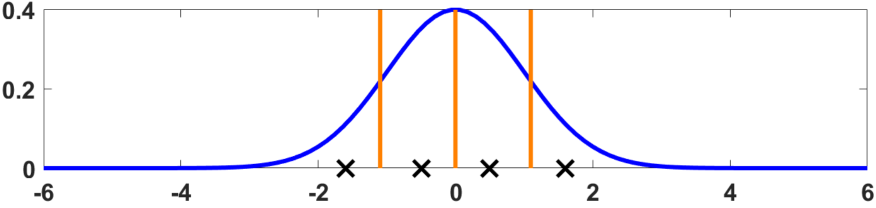
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



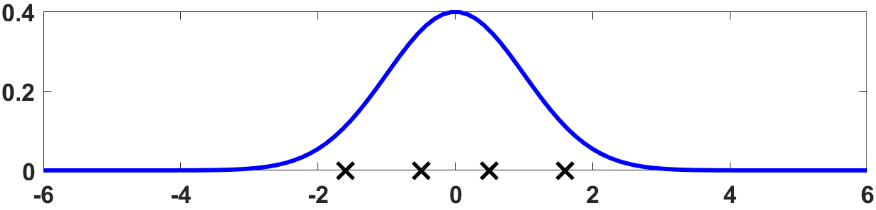
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



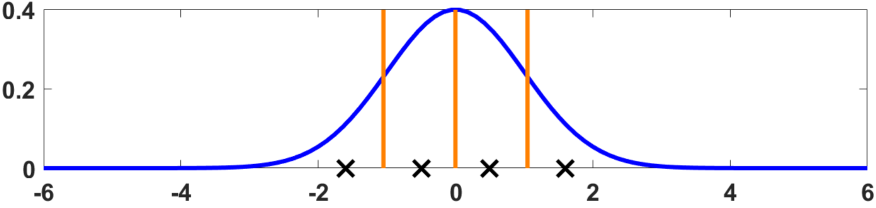
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



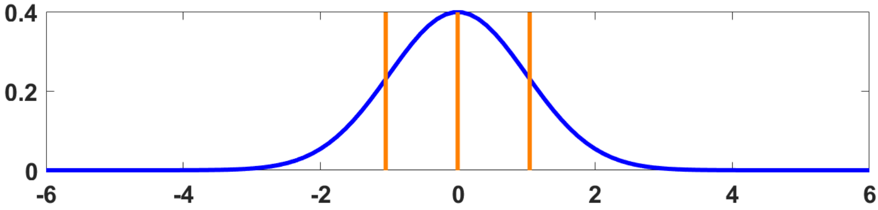
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



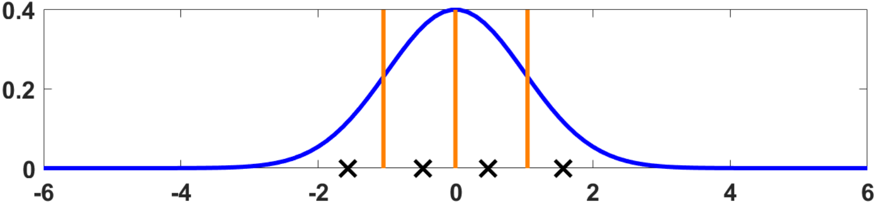
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2,\ \forall j \neq i\right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



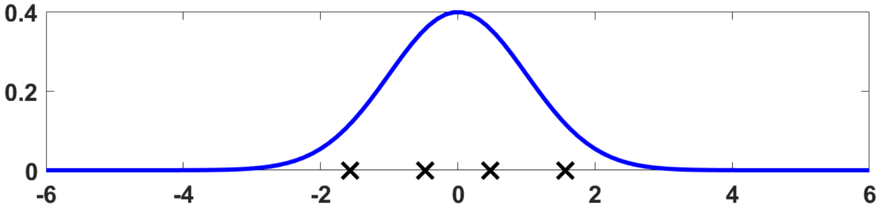
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



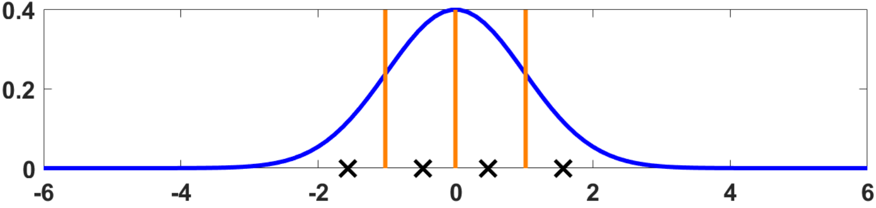
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



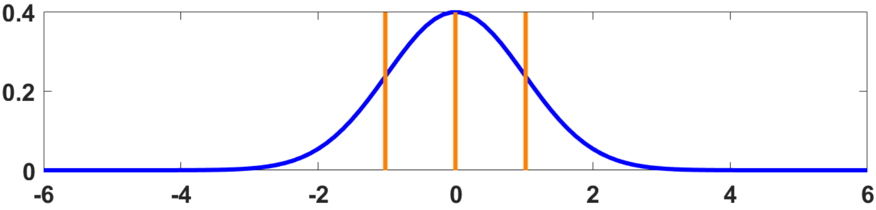
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x-\hat{x}_i)^2 < (x-\hat{x}_j)^2, \ \forall j \neq i\right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



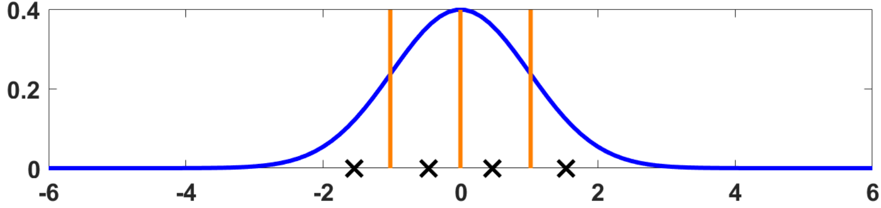
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



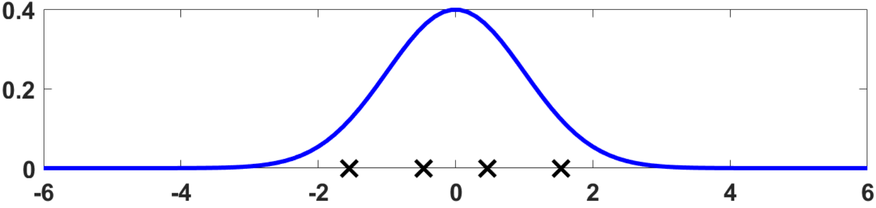
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



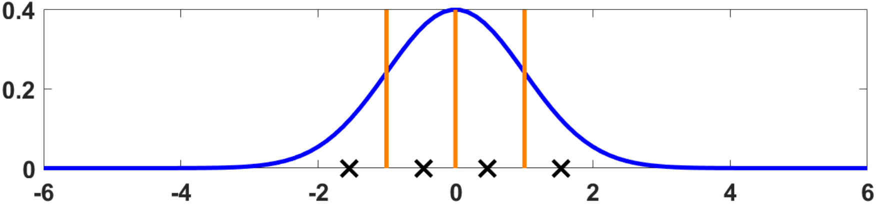
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



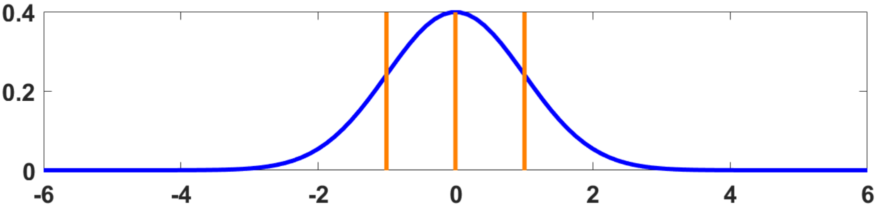
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i\right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



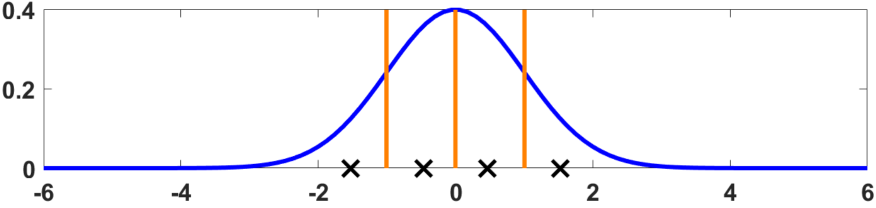
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i\right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2,\ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$

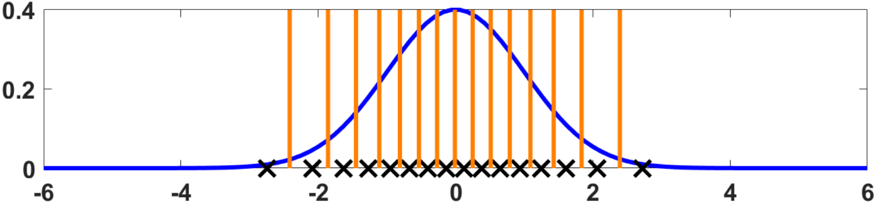
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

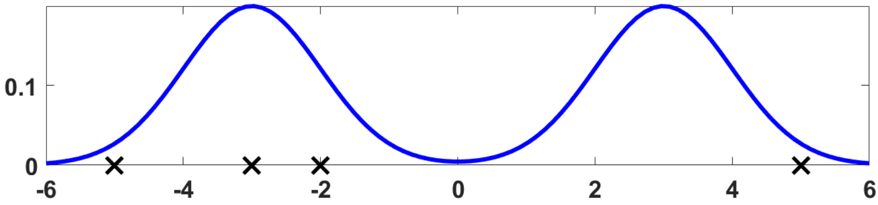
Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



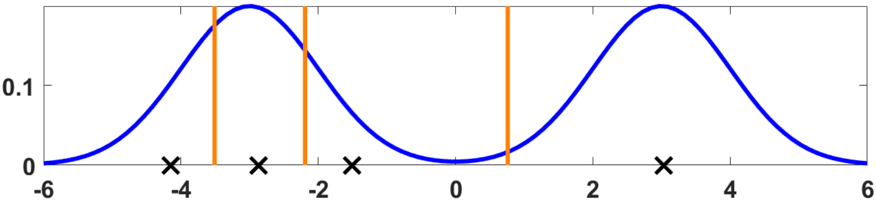
Lloyd-Max Algorithm

- Optimal quantizer necessarily satisfies *Centroid* and *NN*
- But... They are not sufficient in general! ☹
- Lloyd-Max algorithm might converge to a local optimum...



Lloyd-Max Algorithm

- Optimal quantizer necessarily satisfies *Centroid* and *NN*
- But... They are not sufficient in general! ☹
- Lloyd-Max algorithm might converge to a local optimum...



Lloyd-Max Algorithm

When does Lloyd-Max converge to global optimum?
[Fleischer '64][Trushkin IT'82][Kieffer-Jahns-Obuljen IT'88]

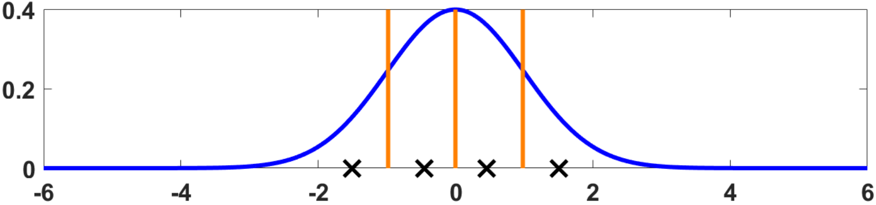
- Conditions for existence of only one local optimum $\Rightarrow$ **Global**
- **Log-concave** distributions satisfy these conditions
- Important special case: **Gaussian distribution** ☺

- One time step of LQG with finite-rate noiseless channel ✓

What about more time step?

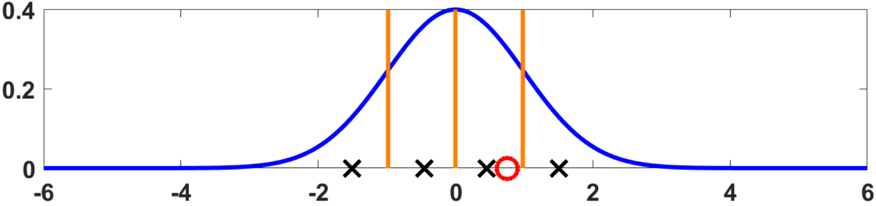
Multi-Step Control with Finite-Rate Feedback [Kh. et al. CDC'17]

- First input $x_1 = w_0$ is Gaussian $\Rightarrow$ Log-concave pdf
- Lloyd-Max quantizer is optimal



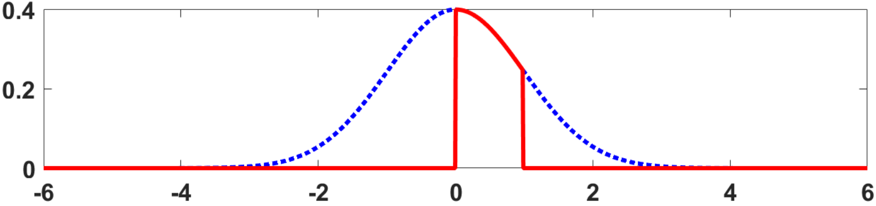
Multi-Step Control with Finite-Rate Feedback [Kh. et al. CDC'17]

- First input x_1 arrives
- Chooses cell: cell i
- Chooses reconstruction point: $\hat{x}_i$



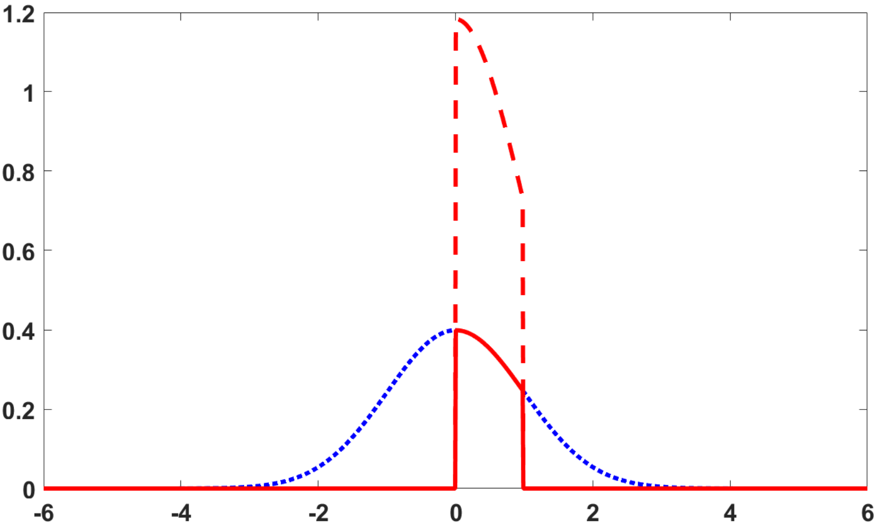
Multi-Step Control with Finite-Rate Feedback [Kh. et al. CDC'17]

- pdf given hit cell i $p(x_1 | x_1 \in \text{cell } i)$ = truncated original pdf $p(x_1)$



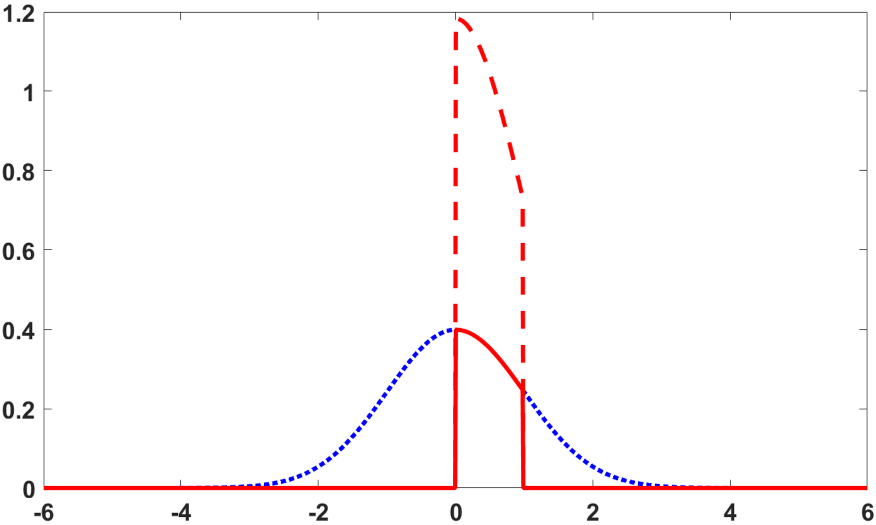
Multi-Step Control with Finite-Rate Feedback [Kh. et al. CDC'17]

- Up to scaling...



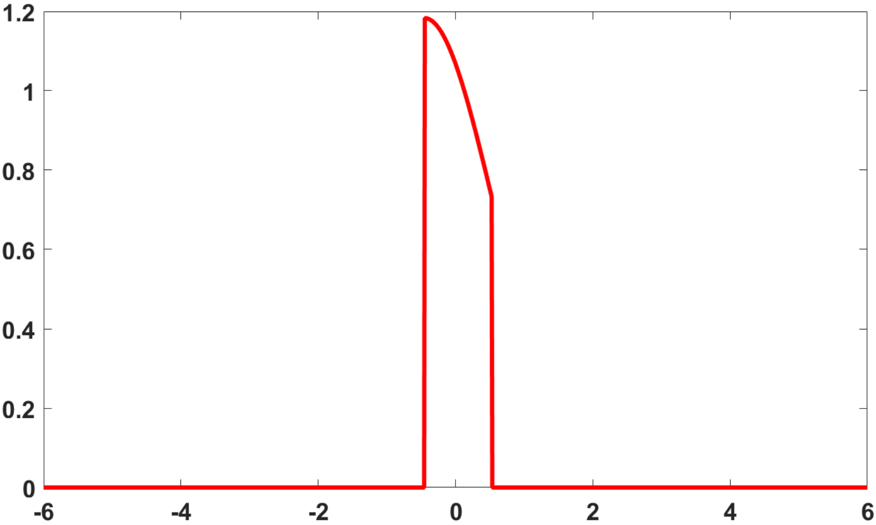
Multi-Step Control with Finite-Rate Feedback [Kh. et al. CDC'17]

- Truncated log-concave pdf is **log-concave**!



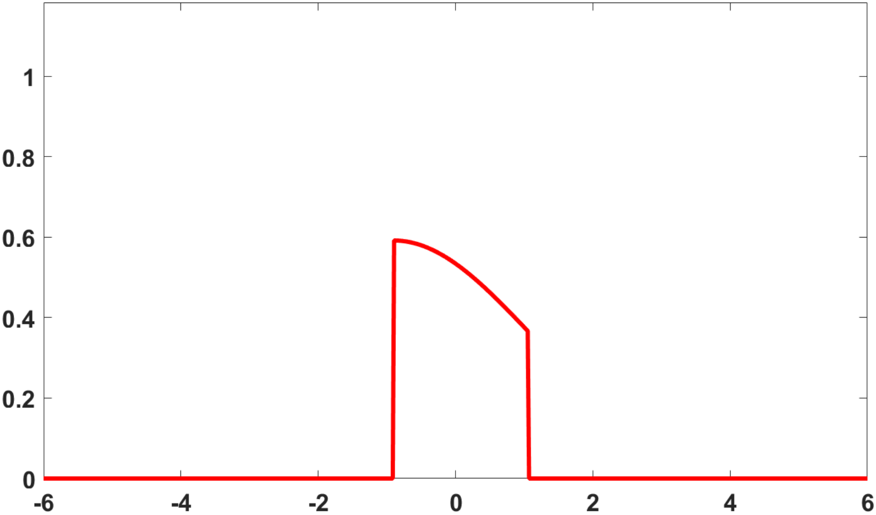
Multi-Step Control with Finite-Rate Feedback [Kh. et al. CDC'17]

- pdf of quantization noise $p(x_1 - \hat{x}_1 | x_1 \in \text{ cell } i)$



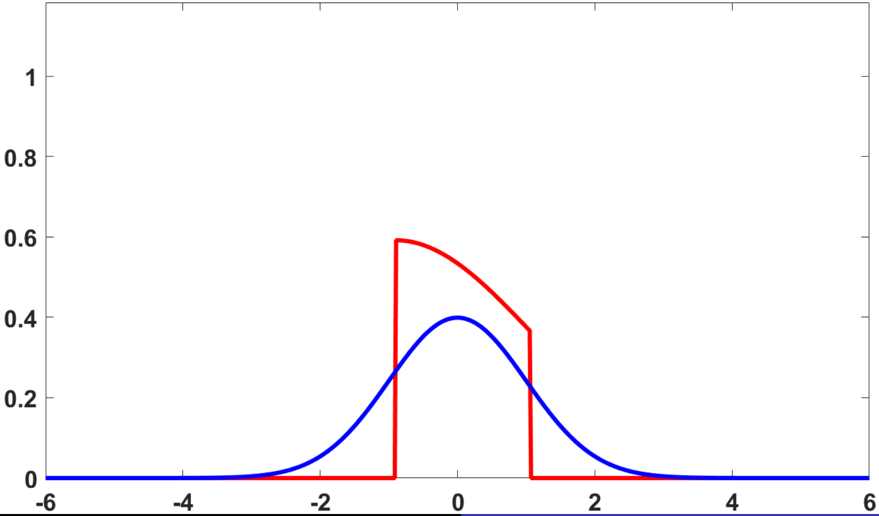
Multi-Step Control with Finite-Rate Feedback [Kh. et al. CDC'17]

- Quantization noise inflated by α: $\alpha(x_1 - \hat{x}_1)$



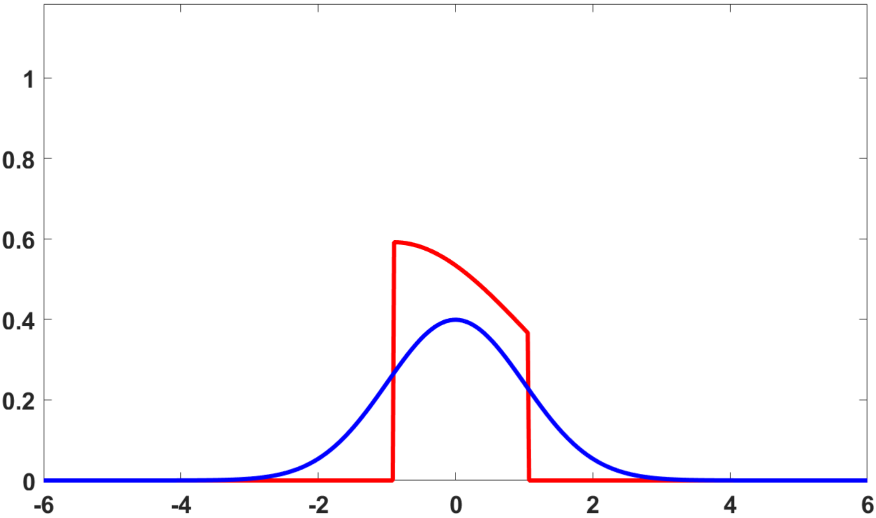
Multi-Step Control with Finite-Rate Feedback [Kh. et al. CDC'17]

- New w_t added: $\alpha(x_1 - \hat{x}_1) + w_1 \Rightarrow$ Convolution of pdfs



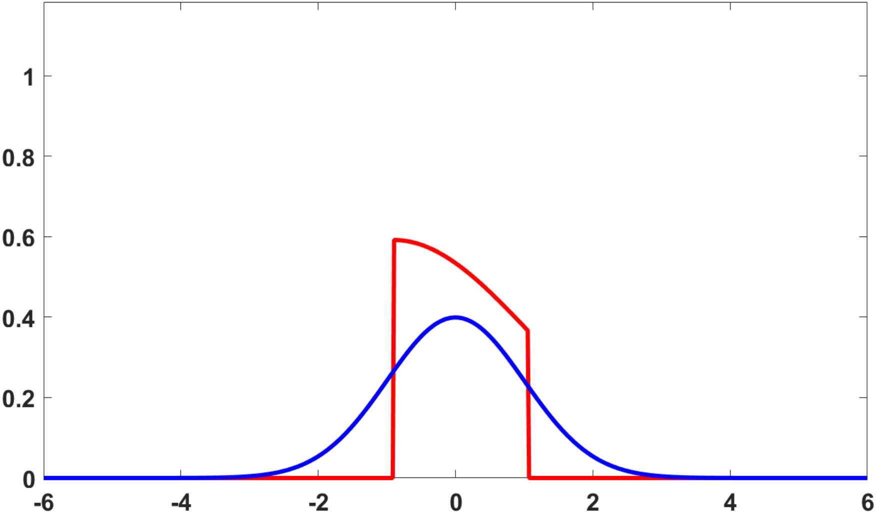
Multi-Step Control with Finite-Rate Feedback [Kh. et al. CDC'17]

- $w_t \sim \mathcal{N}(0, W) \ast$ log-concave quantization error



Multi-Step Control with Finite-Rate Feedback [Kh. et al. CDC'17]

- Convolution of log-concave functions is also **log-concave**!



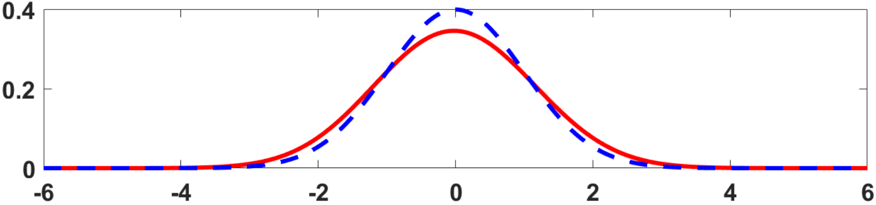
Multi-Step Control with Finite-Rate Feedback [Kh. et al. CDC'17]

Resulting pdf (in red)

- Depends on cell index chosen in previous step(s)
- Log-concave

Applying Lloyd-Max quantization in second step is optimal!

- First-step pdf (in blue) for comparison



Optimal Greedy Algorithm

- Lloyd-Max quantization minimizes squared error of that step
- Lloyd-Max quantization = Optimal **greedy** algorithm
- But... It is not necessarily globally optimal...
- Quantizer used affects pdf of future time steps
- Quantizer should be chosen according to the dynamic program (take into account the "cost-to-go")

Greedy optimal vs. Globally optimal

Low rates: Negligible loss $\sim 1\% - 2\%$

High rates: Can be proved to be optimal via Bennett's rule

Back to the Gaussian Channel Setting

We developed two ingredients:

1. **Tree code** transform the problem:
   Noisy channel $\Rightarrow$ Noiseless channel with random delay
2. **Lloyd–Max-based scheme**
   used over the resulting noiseless channel

Separation-based scheme

Encoder:

- Applies Lloyd-Max-based scheme
- Encodes quantized bits using a tree code

Decoder:

- Recovers all coded bits
- If error is detected $\rightarrow$ rerun LM from that point

Joint Source–Channel Coding

Linear Quadratic Gaussian Control over Gaussian Channels

Control rate $\neq$ Communication rate!

- Assume N channel uses per one control sample

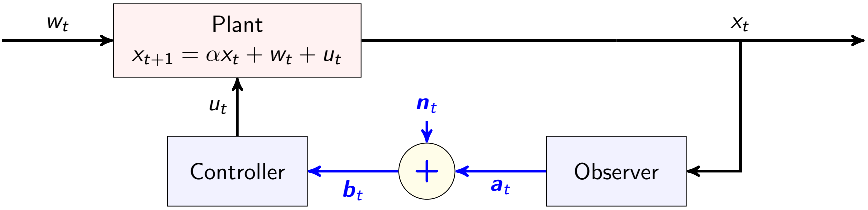
Scalar LQG system

$$x_{t+1} = \alpha x_t + u_t + w_t$$

Scalar AWGN channel

$$\boldsymbol{b}_t = \boldsymbol{a}_t\left(x^t, u^{t-1}\right) + \boldsymbol{n}_t$$

Power constraint: $\mathbb{E}\left[\boldsymbol{a}_t^2\right] \leq NP$



1 : 1 JSCC: Rate-Matched Case

- One AWGN channel use per one control sample

1 : 1 Optimal JSCC [Goblick IT'65]

1 : 1 optimal JSCC distortion = n : n optimal JSCC distortion

- No loss of performance
- **Analog scheme is optimal:** $a_t = \sqrt{\frac{P}{P_x}} x_t$

1 : 1 JSCC: Rate-Matched Case

Scheme

Observer/Transmitter:

- Generates the "source" signal: $s_t = x_t - \hat{x}_{t|t-1} = \tilde{x}_{t|t-1}$
- Adjusts power and transmits: $a_t = s_t / \sqrt{P_{t|t-1}}$

Controller/Receiver:

- Receives $b_t = a_t + n_t = \tilde{x}_{t|t-1} / \sqrt{P_{t|t-1}} + n_t$
- Applies Kalman filtering: $\begin{cases} \hat{x}_{t|t} = \hat{x}_{t|t-1} + \sqrt{P_{t|t-1}} \frac{\mathrm{SNR}}{1+\mathrm{SNR}} b_t \\ \hat{x}_{t|t-1} = \alpha \hat{x}_{t-1|t-1} + u_{t-1} \end{cases}$
- Generates LQG control signal: $u_t = -L_t \hat{x}_{t|t}$

1 : 1 JSCC: Rate-Matched Case

- We reduced the problem to that of classical LQG control

LQR coefficients

$$L_t = \frac{\alpha S_{t+1}}{S_{t+1} + R}\,,$$

$$S_t = \frac{\alpha^2 R\, S_{t+1}}{S_{t+1} + R} + Q\,,$$

$$S_T = F\,.$$

1 : 1 JSCC: Rate-Matched Case

LQG cost

- This schemes achieves optimal LQG cost
- Formally proved by applying
  - Shannon's lower bound
  - Entropy-power inequality
  - Tightness of both in Gaussian case
  - Optimality of "1 : 1 JSCC" scheme in the Gaussian case

  in the dynamic-programming solution
  (extension of [Kostina-Hassibi Allerton'16]
  [Kh. et al. ITW'17 & submitted TCNS'17])
- Recovers results of [Freudenberg-Middleton-Solo AC'10] as a special case

Conclusion: No coding is needed!

1 : 1 JSCC: Rate-Matched Case

Optimal infinite-horizon steady-state average-time LQG cost

$$\bar{J}^{\mathrm{r}} = \bar{J}^{\mathrm{t}} + \frac{Q + \left(\alpha^2 - 1\right) S}{1 + \mathsf{SNR} - \alpha^2} W$$

$$\bar{J}^{\mathrm{t}} = SW$$

- S is the positive solution of the DARE

$$S^2 - \left[Q + \left(\alpha^2 - 1\right) R\right] S - QR = 0$$

- System is stabilizable if and only if $\mathsf{SNR} > \alpha^2 - 1$
- This is in stark contrast to classical LQG

Conclusion: No coding is needed!

1 : 1 JSCC: Rate-Matched Case

Optimal infinite-horizon steady-state average-time LQG cost

$$\bar{J}^{\mathrm{r}} = \bar{J}^{\mathrm{t}} + \frac{Q + \left(\alpha^2 - 1\right) S}{1 + \mathsf{SNR} - \alpha^2} W$$
$$\bar{J}^{\mathrm{t}} = SW$$

- S is the positive solution of the DARE
$$S^2 - \left[Q + \left(\alpha^2 - 1\right) R\right] S - QR = 0$$
- System is stabilizable if and only if $\mathsf{SNR} > \alpha^2 - 1$
- This is in stark contrast to classical LQG

What about 1 : 2 case?

1 : 2 JSCC: Rate-Mismatched Case

- Two AWGN channel uses per one control sample

Naïve scheme: Repetition

Observer/Transmitter: $a_{t;1} = a_{t;2} = \tilde{x}_t / \sqrt{P_{t|t-1}}$

Controller/Receiver: $b_t^{\text{eff}} = \frac{b_{t;1} + b_{t;2}}{2}$

- Reduces to 1 : 1 JSCC with $\text{SNR}^{\text{eff}} = 2\text{SNR}$
- 3dB improvement comes from doubling total transmit power
- Same improvement is attained by
  - Using $2P$ during first channel use
  - Remaining silent during second channel use
- No real improvement due to extra degree of freedom...

1 : 2 JSCC: Rate-Mismatched Case

- Two AWGN channel uses per one control sample

Naïve scheme: Repetition

Observer/Transmitter: $a_{t;1} = a_{t;2} = \tilde{x}_t / \sqrt{P_{t|t-1}}$

Controller/Receiver: $b_t^{\mathrm{eff}} = \frac{b_{t;1}+b_{t;2}}{2}$

- Reduces to 1 : 1 JSCC with $\mathrm{SNR}^{\mathrm{eff}} = 2\mathrm{SNR}$
- 3dB improvement comes from doubling total transmit power
- Same improvement is attained by
  - Using $2P$ during first channel use
  - Remaining silent during second channel use
- No real improvement due to extra degree of freedom...

Can we do better?

1 : 2 JSCC: Rate-Mismatched Case

Infinite blocklength: "n : 2n JSCC" for $n \to \infty$ [Shannon '48]

$$1 + \mathsf{SNR}^{\mathrm{eff}} = (1 + \mathsf{SNR})^2$$

- Much better than $\mathsf{SNR}^{\mathrm{eff}}_{\text{naïve}} = 2\mathsf{SNR}$ at high SNR

1 : 2 JSCC: Rate-Mismatched Case

Infinite blocklength: "n : 2n JSCC" for $n \to \infty$ [Shannon '48]

$$1 + \mathrm{SNR}^{\mathrm{eff}} = (1 + \mathrm{SNR})^2$$

- Much better than $\mathrm{SNR}^{\mathrm{eff}}_{\text{naïve}} = 2\mathrm{SNR}$ at high SNR

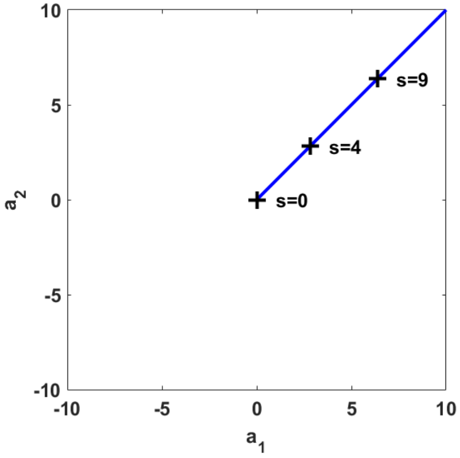
What about 1 : 2 JSCC?

Non-linear mappings can do better! [Kotel'nikov '47][Shannon '49]

1 : 2 JSCC: Rate-Mismatched Case

1 : 2 JSCC: Rate-Mismatched Case

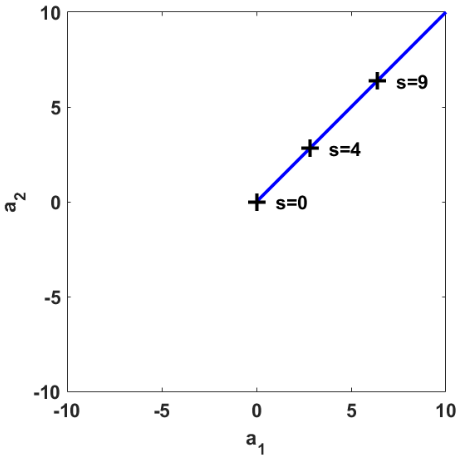
$$\begin{cases} a_1(s) = \frac{1}{\sqrt{2}} s \\ a_2(s) = \frac{1}{\sqrt{2}} s \end{cases}$$

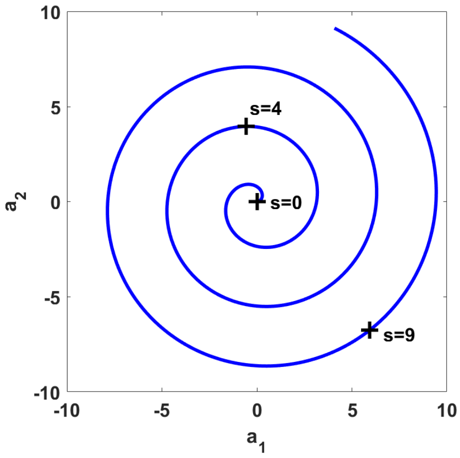


1 : 2 JSCC: Rate-Mismatched Case

$$\begin{cases} a_1(s) = \frac{1}{\sqrt{2}} s \\ a_2(s) = \frac{1}{\sqrt{2}} s \end{cases}$$

$$\begin{cases} a_1(s) = s\cos(2s) \\ a_2(s) = s\sin(2s) \end{cases}$$

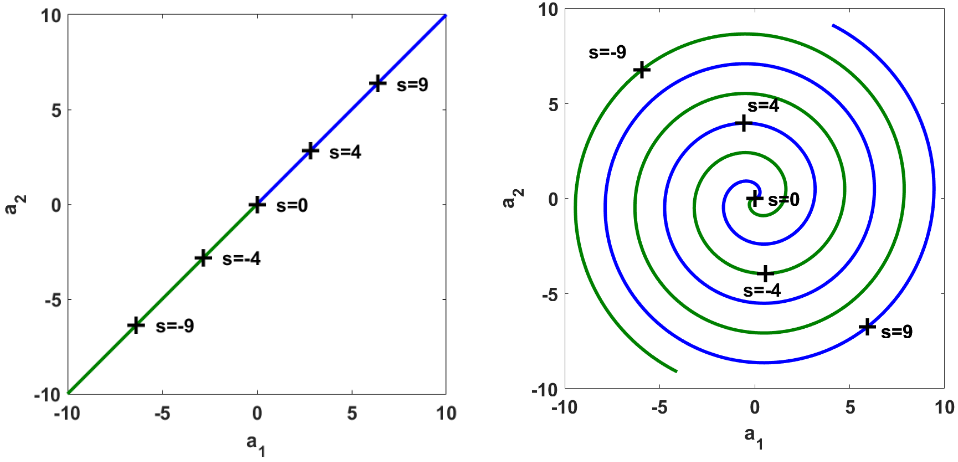




1 : 2 JSCC: Rate-Mismatched Case

$$\begin{cases} a_1(s) = \frac{1}{\sqrt{2}} s \\ a_2(s) = \frac{1}{\sqrt{2}} s \end{cases}$$

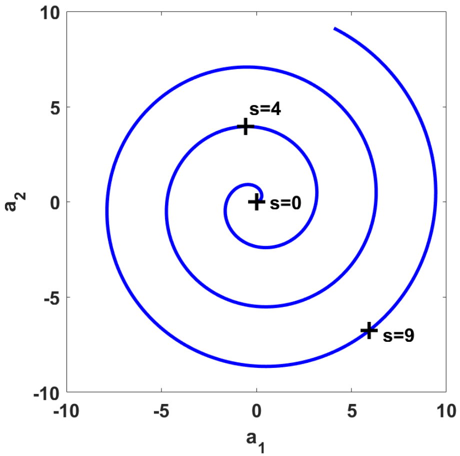
$$\begin{cases} a_1(s) = s\cos(2s) \\ a_2(s) = s\sin(2s)\,\mathrm{sign}(s) \end{cases}$$

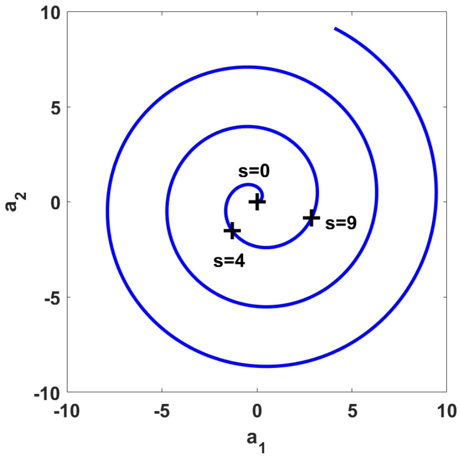


1 : 2 JSCC: Rate-Mismatched Case

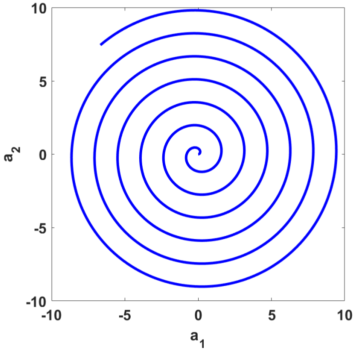
$$\begin{cases} a_1(s) = s\cos(2s) \\ a_2(s) = s\sin(2s) \end{cases}$$

$$\begin{cases} a_1(s) = \sqrt{s}\cos(2\sqrt{s}) \\ a_2(s) = \sqrt{s}\sin(2\sqrt{s}) \end{cases}$$





1 : 2 JSCC: Rate-Mismatched Case

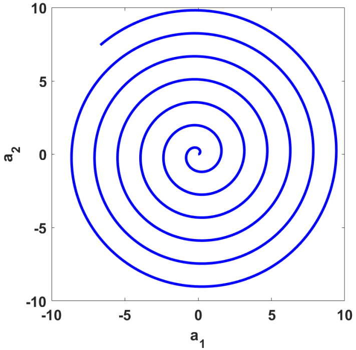


- Small distance between branches
  $\Rightarrow$ better for "weak noise"
- Large distance between branches
  $\Rightarrow$ better for "strong noise"

Standard spiral

$$\begin{cases} a_1(s) \propto s\cos(\omega s) & = |s|\cos(\omega|s|)\,\mathrm{sign}(s) \\ a_2(s) \propto s\sin(\omega s)\,\mathrm{sign}(s) & = |s|\sin(\omega|s|)\,\mathrm{sign}(s) \end{cases}$$

1 : 2 JSCC: Rate-Mismatched Case



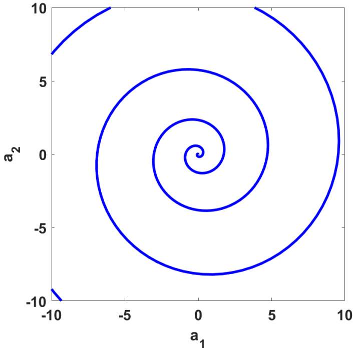
- Small distance between branches
  $\Rightarrow$ better for "weak noise"
- Large distance between branches
  $\Rightarrow$ better for "strong noise"

Stretched-source spiral

Stretch input before mapping to spiral: $s \to |s|^\lambda \operatorname{sign}(s)$

$$\begin{cases} a_1(s) \propto |s|^\lambda \cos\left(\omega |s|^\lambda\right) \operatorname{sign}(s) \\ a_2(s) \propto |s|^\lambda \sin\left(\omega |s|^\lambda\right) \operatorname{sign}(s) \end{cases}$$

1 : 2 JSCC: Rate-Mismatched Case



⟸ Control requirements

- Small distance between branches
  ⇒ better for "weak noise"
- Large distance between branches
  ⇒ better for "strong noise"

Bounded average distortion given any input

Avoid increase in distortion with $|s|$ ⇒ Slower rotation with $|s|$

$$\begin{cases} a_1(s) \propto |s|^{\lambda\beta} \cos\left(\omega|s|^{\lambda}\right) \operatorname{sign}(s) \\ a_2(s) \propto |s|^{\lambda\beta} \sin\left(\omega|s|^{\lambda}\right) \operatorname{sign}(s) \end{cases}$$

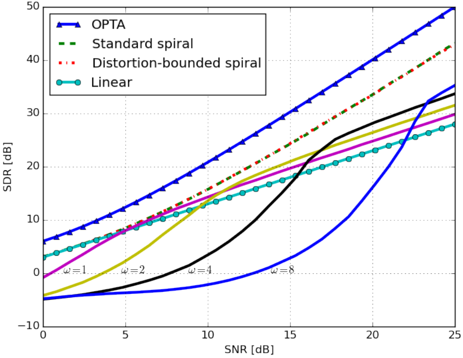
1 : 2 JSCC: Rate-Mismatched Case

⟸ Control requirements

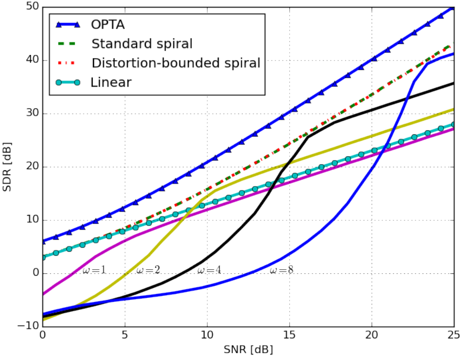
- Small distance between branches
  ⇒ better for "weak noise"
- Large distance between branches
  ⇒ better for "strong noise"

1 : 2 JSCC: Rate-Mismatched Case

- Average distortion given (almost) any s needs to be small!
- E.g., transmitters that truncate the signal do not perform well (avalanche effect)



(a) $\lambda = 1$.



(b) $\lambda = 0.5$.

1 : 2 JSCC: Rate-Mismatched Case

[Kh.-Riedel Gårding-Pettersson-Kostina-Hassibi CDC'16, submitted AC'17]

Inner bound: Black-box approach

Assume a JSCC scheme with bounded distortion $D = \frac{1}{\mathrm{SNR}^{\mathrm{eff}}}, \forall s$.

$$\bar{J}^{\mathrm{r}} \leq \bar{J}^{\mathrm{t}} + \frac{Q + (\alpha^2 - 1)\,S}{1 + \mathrm{SNR}^{\mathrm{eff}} - \alpha^2}(P_t^t - \bar{P}_t^t)$$

- **Improved stabilizability:** $\mathrm{SNR}^{\mathrm{eff}} \geq \alpha^2 - 1$

Outer bound: Extension of [Kostina-Hassibi Allerton'16], [Kh.-Kostina-Khisti-Hassibi ITW'17, submitted TCNS'17]

$$\bar{J}^{\mathrm{r}} \geq \bar{J}^{\mathrm{t}} + \frac{Q + (\alpha^2 - 1)\,S}{1 + \mathrm{SNR}^{\mathrm{eff}}_{n\to\infty} - \alpha^2}(P_t^t - \bar{P}_t^t)$$

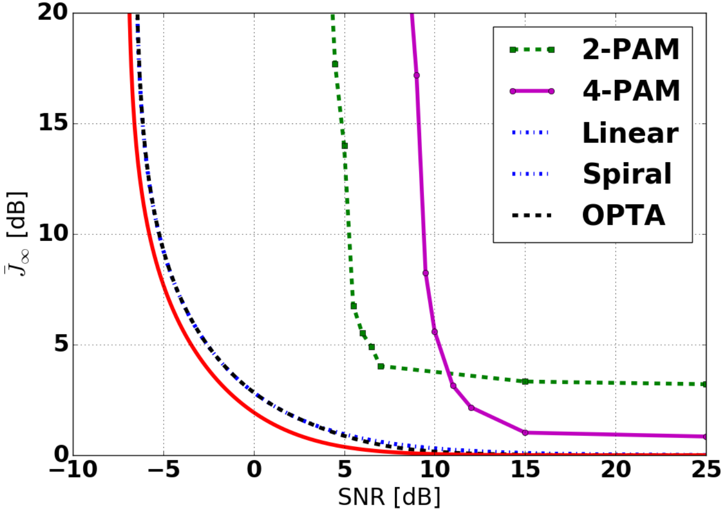
- $1 + \mathrm{SNR}^{\mathrm{eff}}_{n\to\infty} = (1 + \mathrm{SNR})^2$

- Difference between bounds is only due to effective SNR

Performance Comparison

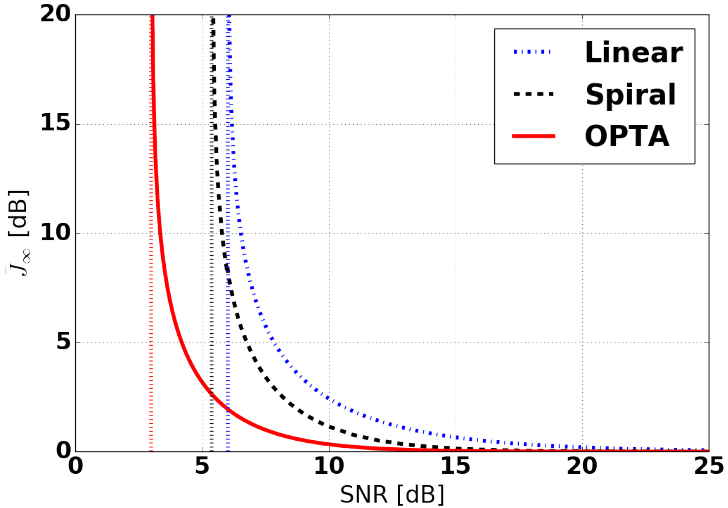
Performance Comparison

- $\alpha = 1.2, W_1 = 1, Q_t \equiv 1, R_t = 0$



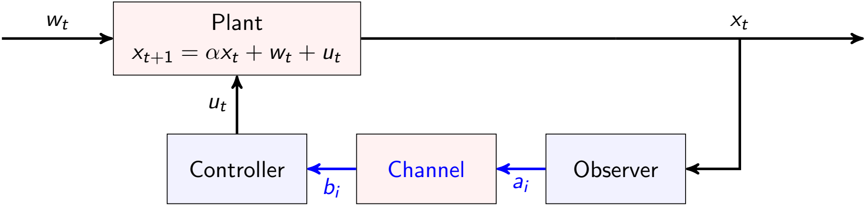
Performance Comparison

- $\alpha = 3, W_1 = 1, Q_t \equiv 1, R_t = 0$



Further Down the Road...

Outlook: Control Loop as Implicit Channel Feedback



- Control signal u_t is assumed known at observer/transmitter
- Can be used as implicit channel feedback
- Could be noisy
- Special instance: JSCC schemes over the AWGN channel

Outlook: Control Loop as Implicit Channel Feedback

Posterior Matching (PM) [Shayevitz-Feder IT'11, IT'16] [Li-El Gamal IT'15][Naghshvar-Javidi-Wigger IT'15], ...

- Fits an initial state (LQR) setting: $\begin{cases} x_{t+1} = \alpha x_t + u_t + \cancel{w_t} \\ x_0 \sim \text{random} \end{cases}$
- Assumes perfect instantaneous feedback is available
- Upon receiving channel output b_t recalculates posterior:

$$\Theta_0 = F_{X_0}(X_0), \quad A_1 = F_A^{-1}(\Theta_0)$$
$$A_{t+1} = F_A^{-1} \circ F_{\Theta_0|B^t}(\Theta_0|b^t)$$

- Transmitter knows b^t via feedback
- Can be calculated iteratively:

$$A_1 = F_A^{-1} \circ F_{X_0}(X_0)$$
$$A_{t+1} = F_A^{-1} \circ F_{A|B}(A_t|b_t)$$

Outlook: Control Loop as Implicit Channel Feedback

Posterior Matching (PM) [Shayevitz-Feder IT'11, IT'16] [Li-El Gamal IT'15][Naghshvar-Javidi-Wigger IT'15], ...

- Fits an initial state (LQR) setting: $\begin{cases} x_{t+1} = \alpha x_t + u_t + \cancel{w_t} \\ x_0 \sim \text{random} \end{cases}$
- Assumes perfect instantaneous feedback is available
- Upon receiving channel output b_t recalculates posterior:

$$A_1 = F_A^{-1} \circ F_{X_0}(X_0)$$
$$A_{t+1} = F_A^{-1} \circ F_{A|B}(A_t|b_t)$$

Problems

1. What to do for i.i.d. w_t? $\Rightarrow$ Causal variant of PM is needed!
2. We are interested in control-theoretic notions, say LQG cost

Outlook: Control Loop as Implicit Channel Feedback

BSC: Horstein's scheme [Horstein IT'63]

- Special case of PM scheme over the BSC
- At every step:
  - Calculates posterior
  - Sends whether the posterior of $X_0 \lessgtr$ Median

- Not bad for first moment minimization $\sum_{t=1}^{T} \mathbb{E}\left[\left|X_0 - \hat{X}_0(t)\right|\right]$
- Analysis (not tight!) in [Waeber-Frazier-Henderson SICON'13]
- What about LQG cost, say $\sum_{t=1}^{T} \mathbb{E}\left[\left|X_0 - \hat{X}_0(t)\right|^2\right]$?

Idea: Extend PM scheme to a wider class

- Compare to MMSE instead of median

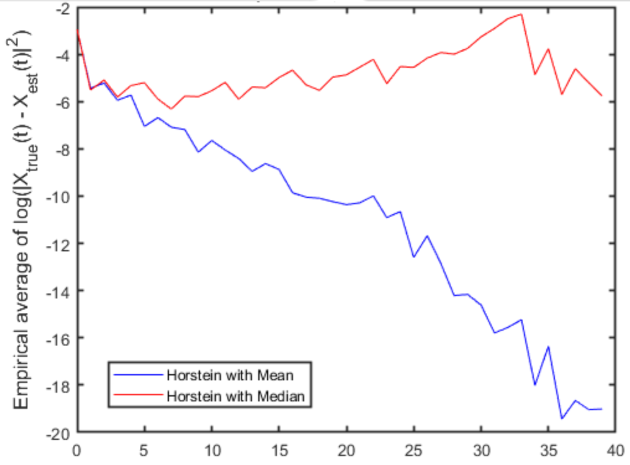
Performance of the Median- and MMSE-based Schemes

Plant

- $x_t = \alpha x_t + u_t + \cancel{w_t}$
- $\alpha = 1.2, x_0 \sim \mathcal{Unif}(0,1)$

Channel

- Binary Symmetric Channel
- Crossover probability 0.025



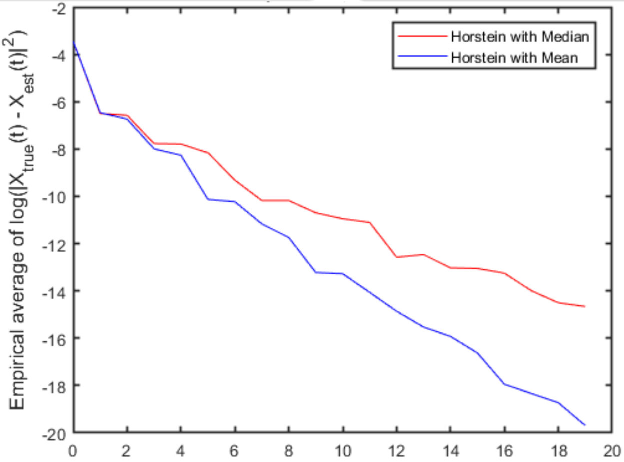
Performance of the Median- and MMSE-based Schemes

Plant

- $x_t = \alpha x_t + u_t + \cancel{w_t}$
- $\alpha = 1.01, x_0 \sim \mathcal{U}nif(0,1)$

Channel

- Binary Symmetric Channel
- Crossover probability 0.025



Collaborators

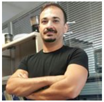
Prof. Uri Erez
Tel Aviv U.

Sergio Escobar
Caltech

Dr. Wael Halbawi
Oracle (Caltech)

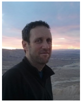
Prof. Babak Hassibi
Caltech

Ayal Hitron
Istra (Tel Aviv U.)



Mustafa Kesal
ASELSAN (Bilkent)

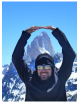
Prof. Ashish Khisti
U. of Toronto



Prof. Yuval Kochman
Hebrew U. (MIT, TAU)

Prof. Victoria Kostina
Caltech

Dr. Moshe Laifenfeld
Apple (GM)



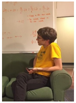
Idan Livni
Raft (Tel Aviv U.)

Yorie Nakahira
Caltech

Dr. Or Ordentlich
Hebrew U. (Tel Aviv U.)

Gustav M. Pettersson
KTH

Dr. Tal Philosof
Samsung (GM, TAU)



Elias Riedel Gårding
Cambridge (KTH)

Yu Su
Caltech

Prof. Gregory W. Wornell
MIT

Dr. Yair Yona
Intel (ULCA, TAU)

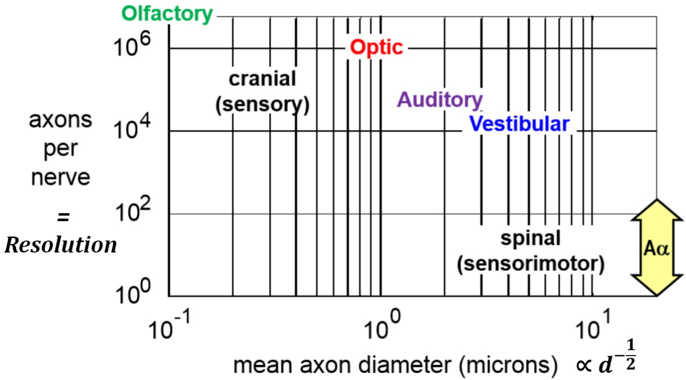
Prof. Ram Zamir
Tel Aviv U.

No relationships were ruined in the making of this presentation...

Backup Slides

Neuroscience: Micro-level [Perge et al. J. Neuroscience'12]

- Cranial and spinal nerves = bundles of fibers (=axons)
- Nerves connect collection of Neurons over long distances
- Have (roughly) the same cross sectional area



[Nakahira et al. CDC'15]

Ensemble Performance ⇒ Specific Code Performance?

Ensemble performance

$$\mathbb{E}\left[P_e(t,d)\right] \le 2^{-E_G(R)Nd}$$

Specific d and t

Using Markov's inequality:

$$\Pr\left(P_e(t,d) \ge 2^{-[E_G(R)-\epsilon]Nd}\right) \le \frac{\mathbb{E}\left[P_e(t,d)\right]}{2^{-[E_G(R)-\epsilon]Nd}} = 2^{-\epsilon Nd}$$

Ensemble Performance ⇒ Specific Code Performance?

Ensemble performance

$$\mathbb{E}\left[P_e(t,d)\right] \leq 2^{-E_G(R)Nd}$$

Specific d and t

Using Markov's inequality:

$$\Pr\left(P_e(t,d) \geq 2^{-[E_G(R)-\epsilon]Nd}\right) \leq \frac{\mathbb{E}\left[P_e(t,d)\right]}{2^{-[E_G(R)-\epsilon]Nd}} = 2^{-\epsilon Nd}$$

All t and $d_0 < d \leq t$

Using the union bound:

$$\Pr\left(\bigcup_{t=1}^{\infty}\bigcup_{d=d_0}^{t} P_e(t,d) \geq 2^{-[E_G(R)-\epsilon]Nd}\right)$$

$$\leq \sum_{t=1}^{\infty}\sum_{d=d_0}^{t} \underbrace{\Pr\left(P_e(t,d) \geq 2^{-[E_G(R)-\epsilon]Nd}\right)}_{\leq 2^{-\epsilon Nd}} \leq \sum_{t=1}^{\infty} \text{const} \to \infty$$

Ensemble Performance ⇒ Specific Code Performance?

Ensemble performance

$$\mathbb{E}\left[P_e(t,d)\right] \le 2^{-E_G(R)Nd}$$

Specific d and t

Using Markov's inequality:

$$\Pr\left(P_e(t,d) \ge 2^{-[E_G(R)-\epsilon]Nd}\right) \le \frac{\mathbb{E}\left[P_e(t,d)\right]}{2^{-[E_G(R)-\epsilon]Nd}} = 2^{-\epsilon Nd}$$

All t and $d_0 < d \le t$

Using the union bound:

$$\Pr\left(\cancel{\bigcup_{t=1}^{\infty}} \bigcup_{d=d_0}^{\cancel{t}\,\to\,\infty} P_e(\cancel{t},d) \ge 2^{-[E_G(R)-\epsilon]Nd}\right)$$

$$\le \cancel{\sum_{t=1}^{\infty}} \sum_{d=d_0}^{\cancel{t}\,\to\,\infty} \underbrace{\Pr\left(P_e(\cancel{t},d) \ge 2^{-[E_G(R)-\epsilon]Nd}\right)}_{\le 2^{-\epsilon Nd}} \le \cancel{\sum_{t=1}^{\infty}} \text{const} \to \cancel{\infty}$$

Ensemble Performance ⇒ Specific Code Performance?

Ensemble performance

$$\mathbb{E}\left[P_e(t,d)\right] \le 2^{-E_G(R)Nd}$$

Specific d and t

Using Markov's inequality:

$$\Pr\left(P_e(t,d) \ge 2^{-[E_G(R)-\epsilon]Nd}\right) \le \frac{\mathbb{E}\left[P_e(t,d)\right]}{2^{-[E_G(R)-\epsilon]Nd}} = 2^{-\epsilon Nd}$$

All t and $d_0 < d \le t$

Using the union bound:

$$\Pr\left(\bigcup_{d=d_0}^{\infty} P_e(d) \ge 2^{-[E_G(R)-\epsilon]Nd}\right) \le \sum_{d=d_0}^{\infty} 2^{-\epsilon Nd} = \frac{2^{-\epsilon Nd_0}}{1-2^{-\epsilon N}}$$

Analog Codes / JSCC: Further Results and Comments

- Inner bound can be improved: Optimization over curves, e.g. [Akyol-Vishwanatha-Rose-Ramstad IT'14]
- Outer bound for low-delay JSCC can be improved [Ziv-Zakai IT'73]
- High dimensional curves
- Other low-delay JSCC techniques: e.g., repetitive quantization [Kleiner-Rimoldi GLOBECOM'09]
  - Easy to generalize to higher dimensions
- Vector $\boldsymbol{x}$, vector $\boldsymbol{u}$, scalar y: *Simple extension of scalar setting!*
- Rate-matched case with vector $\boldsymbol{y}$: "n : 1 JSCC" is needed
  - Switch roles between Transmitter and Receiver
  - Improves over [Freudenberg-Middleton-Solo AC'10]