多种随机丢包模型下的迭代学习控制

设计与分析框架

沈栋

2018.03.30

Overview

背景介绍 → **数据丢包** → **相似问题拓展**

背景介绍	数据丢包	相似问题拓展
迭代学习控制	数据丢包模型	复杂通信问题
随机逼近	迭代学习控制问题	随机变长度问题
	设计与分析	

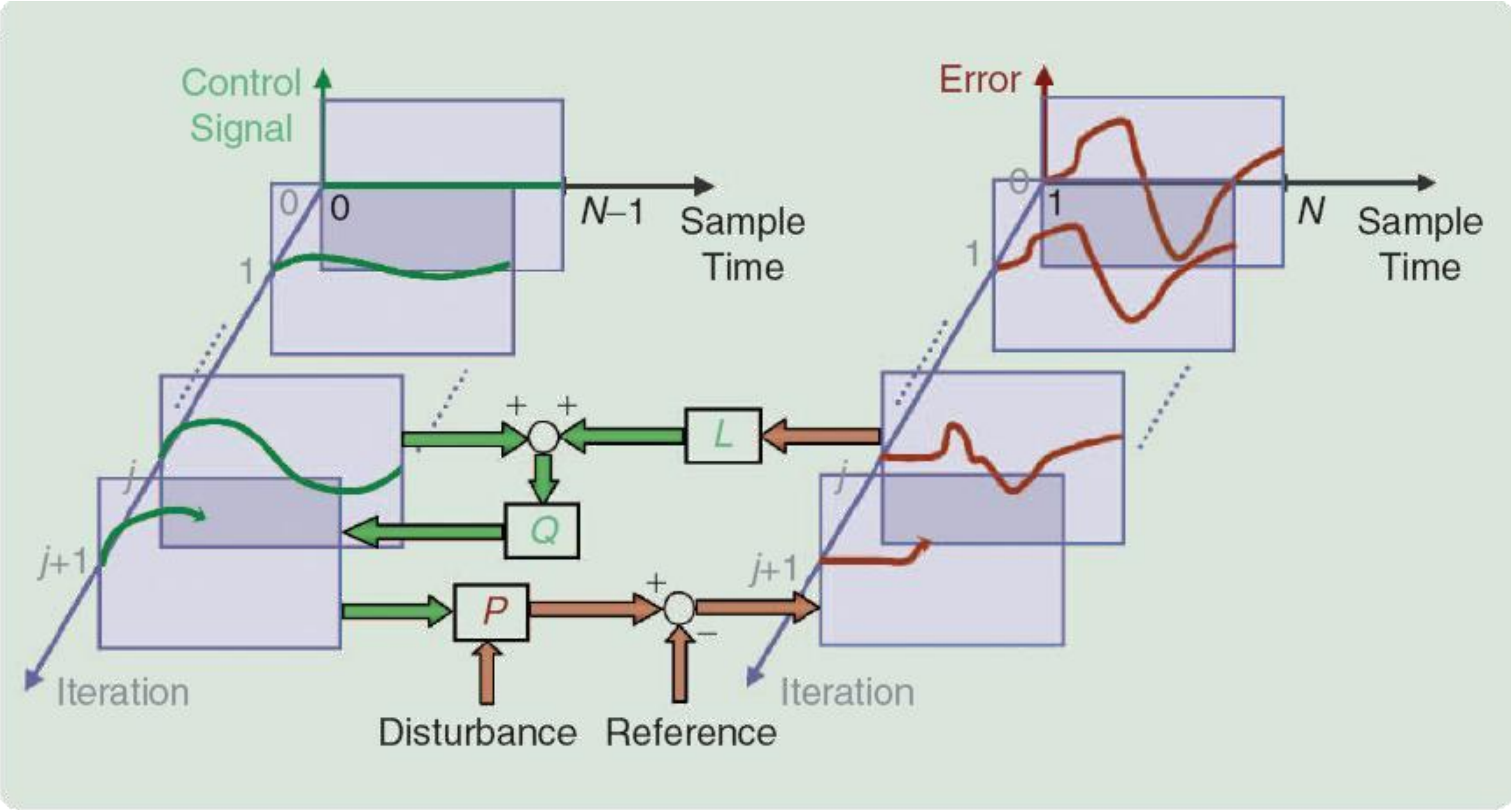
Overview

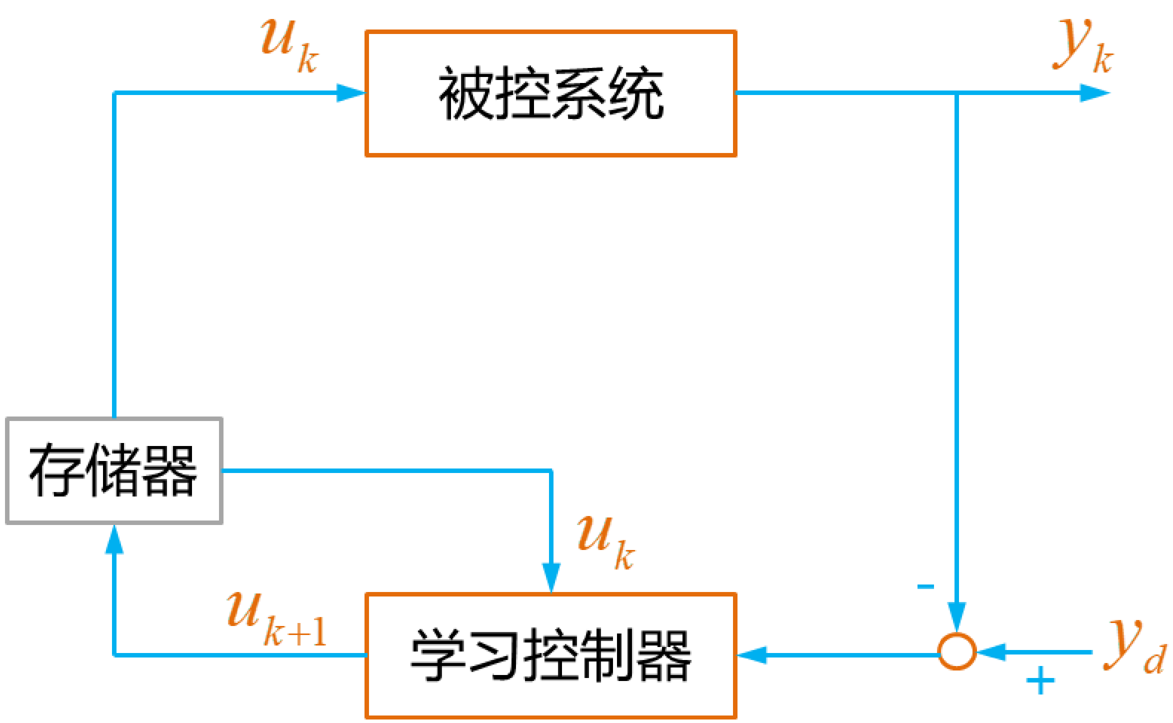
背景介绍 → **数据丢包** → **相似问题拓展**

背景介绍：
- 迭代学习控制
- 随机逼近

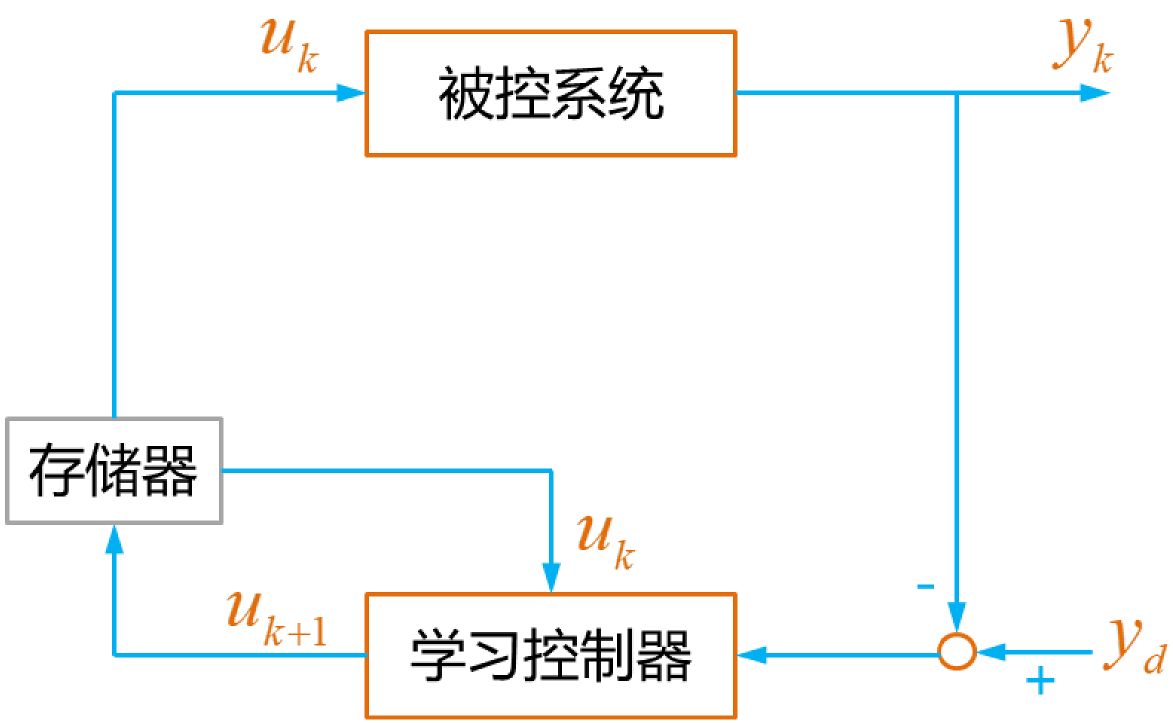
数据丢包：
- 数据丢包模型
- 迭代学习控制问题
- 设计与分析

相似问题拓展：
- 复杂通信问题
- 随机变长度问题


Control Signal
0
0
N–1
Sample Time
1
j
j+1
Iteration
+
+
L
Q
P
+
–
Disturbance
Reference
Error
0
1
N
Sample Time
1
j
j+1
Iteration



u_k
被控系统
y_k
存储器
u_k
u_{k+1}
学习控制器
-
+
y_d



几个特点

系统运行时间有限

系统运行初值重置

重复跟踪同一目标

随机逼近：一种基于含噪声观测值的函数迭代求根办法。

$$\boxed{f(x) = 0}$$

$$y_{k+1} = f(x_k) + \epsilon_{k+1}$$

$$x_{k+1} = x_k + a_k y_{k+1}$$

Han-Fu Chen. Stochastic Approximation and Its Applications. Kluwer Academic Publishers, 2002.

$$y_{k+1} = f(x_k) + \epsilon_{k+1}$$

$$x_{k+1} = x_k + a_k y_{k+1}$$

A 2.2. 存在连续可微函数$v(\cdot) : R^p \to R$使得对任意$\Delta > \delta > 0$,

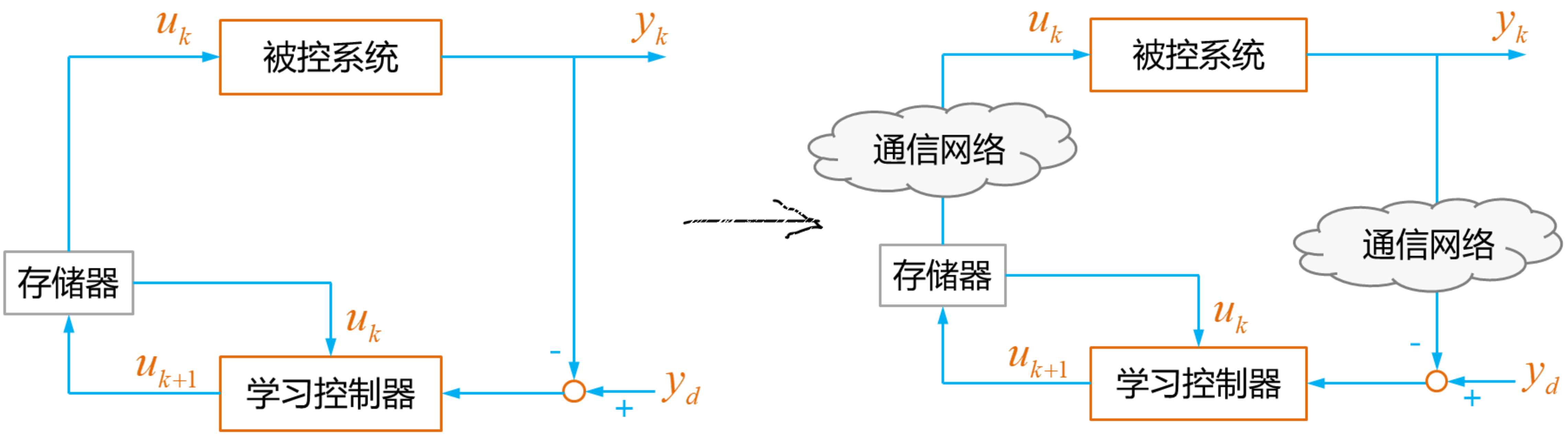
$$\sup_{\delta \le d(x,J) \le \Delta} f^T(x) \frac{\partial v(x)}{\partial x} < 0$$

其中$d(x, J) = \inf_y \|x - y\|, y \in J$, 且$v(J)$无处稠密. 还要求存在$c_0 > 0$使得$\|x^*\| < c_0$且$v(x^*) < \inf_{\|x\|=c_0} v(x)$.

Overview

背景介绍 → **数据丢包** → 相似问题拓展

背景介绍	数据丢包	相似问题拓展
迭代学习控制	数据丢包模型	复杂通信问题
随机逼近	迭代学习控制问题	随机变长度问题
	设计与分析	



u_k
被控系统
y_k
存储器
u_k
u_{k+1}
学习控制器
-
+
y_d
u_k
被控系统
y_k
通信网络
通信网络
存储器
u_k
u_{k+1}
学习控制器
-
+
y_d

数据丢包只有两个状态：丢、不丢

$\gamma_k(t)$	0	1

❶ 伯努利变量模型
BVM

$$\mathbb{P}(\gamma_k(t) = \mathbf{1}) = \gamma$$

$$\mathbb{P}(\gamma_k(t) = \mathbf{0}) = 1 - \gamma$$

$$\gamma = \mathbb{E}\gamma_k(t)$$

$$0 < \gamma < 1$$

独立性

❷ 马氏链模型
MCM

	1	0	

$$P=\begin{bmatrix}P_{11} & P_{10}\\ P_{01} & P_{00}\end{bmatrix}=\begin{bmatrix}\mu & 1-\mu\\ 1-\nu & \nu\end{bmatrix}\begin{matrix}\mathbf{1}\\ \mathbf{0}\end{matrix}$$

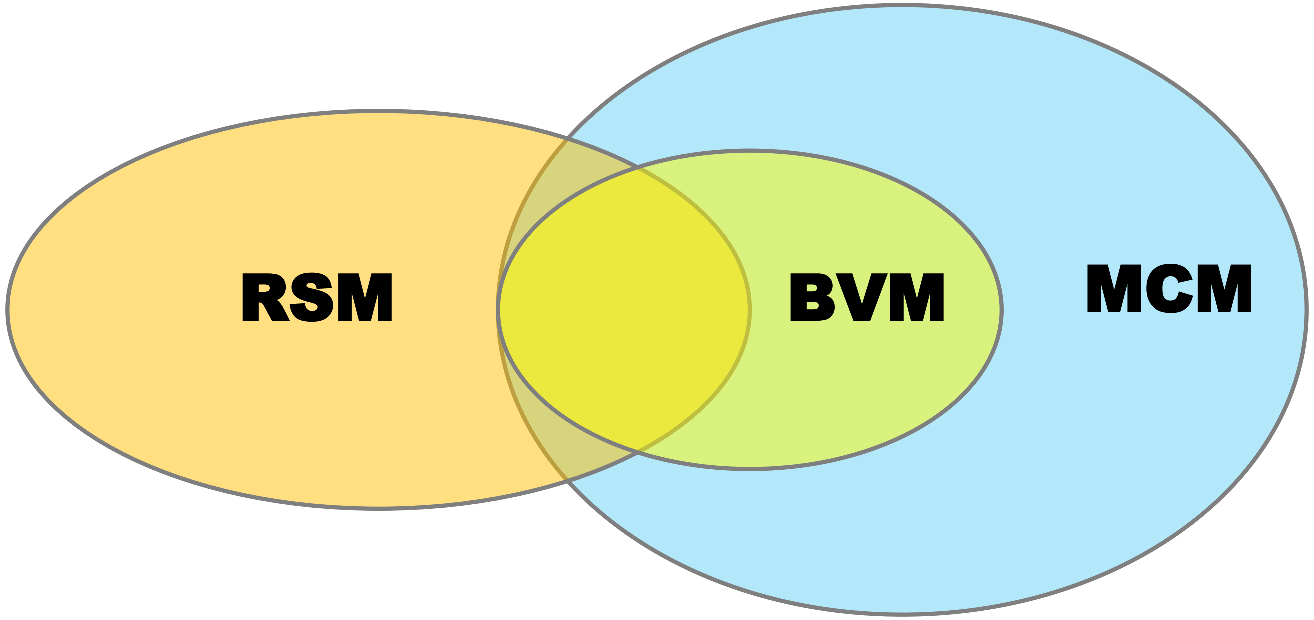
一步依赖性

❸ 随机序列模型
RSM

$$\exists K \quad s.t. \quad \sum_{i=0}^{K-1} \gamma_{k+i}(t) \geq 1$$

消除独立性

相互关系



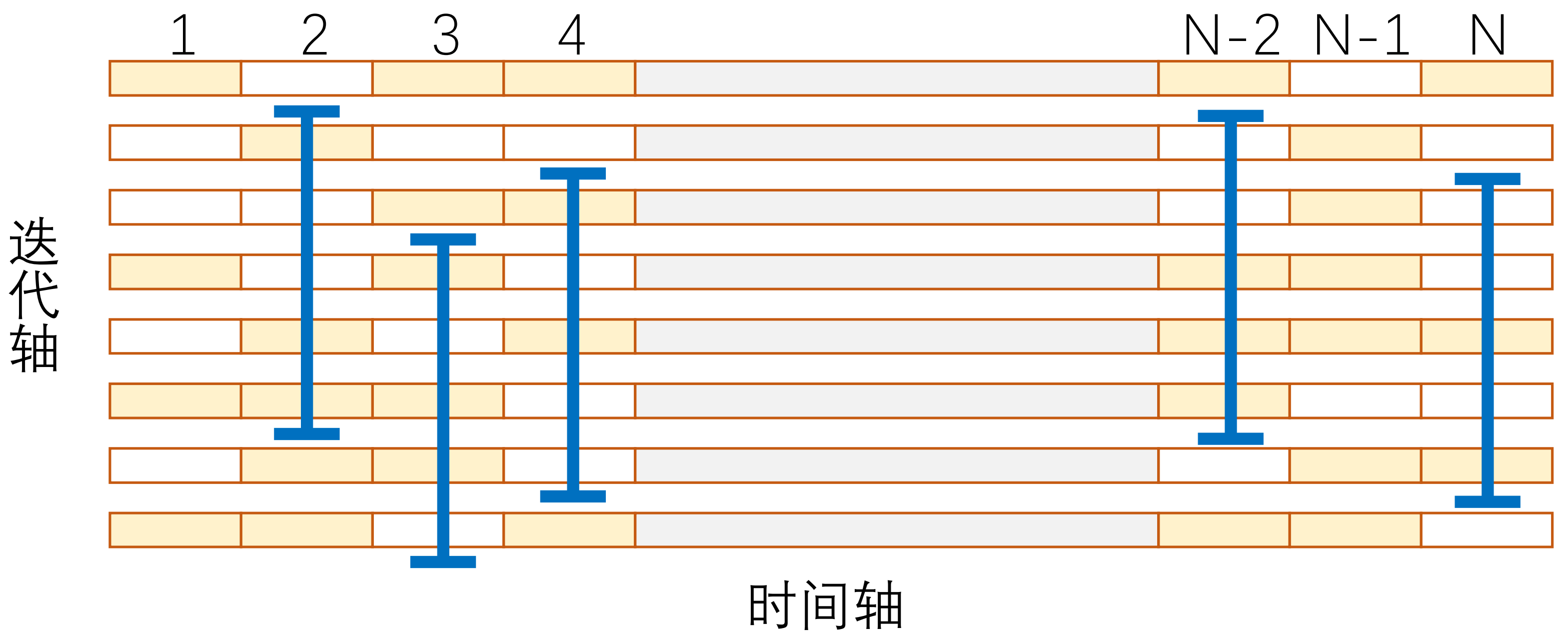
丢包率

Data Dropout Rate

$$\lim_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n}\left(1-\gamma_k(t)\right)$$

BVM	MCM	RSM
$1-\gamma$	$\frac{1-\mu}{2-\mu-\nu}$	

我们从随机序列模型开始



系统

$$x_k(t+1) = f(t, x_k(t)) + \mathbf{b}(t, x_k(t))u_k(t)$$
$$y_k(t) = \mathbf{c}(t)x_k(t) + v_k(t)$$

算法

$$u_{k+1}(t) = u_k(t) + a_k I_{\{(t+1)\notin \mathcal{M}_k\}} \times (y_d(t+1) - y_k(t+1))$$

$$\delta u_{k+1}(t) = \delta u_k(t) - a_k I_{\{(t+1)\notin \mathcal{M}_k\}}[\mathbf{c}^+\mathbf{b}_k(t)\delta u_k(t) + \varphi_k(t) - v_k(t+1)]$$

$$\varphi_k(t) = \mathbf{c}^+\delta f_k(t) + \mathbf{c}^+\delta \mathbf{b}_k(t)u_d(t)$$

$$\delta u_{k+1}(0) = (1 - a_k I_{\{1\notin \mathcal{M}_k\}}\mathbf{c}^+\mathbf{b}_k(0))\delta u_k(0) - a_k I_{1\notin \mathcal{M}_k}\varphi_k(0) + a_k I_{1\notin \mathcal{M}_k} v_k(1)$$

$$\sum_{k=i}^{i+K-1}\left(-I_{\{1\notin \mathcal{M}_k\}}\mathbf{c}^+\mathbf{b}_k(0)\right) < -\gamma, \quad \gamma > 0$$

Dong Shen, Youqing Wang. International Journal of Control, vol. 88, no. 5, pp. 959-968, 2015.

系统

$$x_k(t+1) = f(t, x_k(t)) + \mathbf{b}(t, x_k(t))u_k(t)$$
$$y_k(t) = \mathbf{c}(t)x_k(t) + v_k(t)$$

算法

$$\bar{u}_{k+1}(t) = u_k(t) + a_k(-1)^{\sigma_k(t)} I_{\{(t+1)\notin\mathcal{M}_k\}}(y_d(t+1) - y_k(t+1))$$
$$u_{k+1} = \bar{u}_{k+1} I_{\{|\bar{u}_{k+1}|<M_{\sigma_k(t)}\}}$$
$$\sigma_k(t) = \sum_{j=1}^{k-1} I_{\{|\bar{u}_{j+1}|>M_{\sigma_j(t)}\}}, \qquad \sigma_0(t) = 0$$

方向错误时必发散

方向正确时必有界

Dong Shen, Youqing Wang. System & Control Letters, vol. 77, pp. 30-39, 2015.

下一步，重新考虑伯努利变量模型

$$u_{k+1}(t) = u_k(t) + a_k \gamma_k(t+1) L_t e_k(t+1)$$

$$\delta u_{k+1}(t) = \delta u_k(t) - a_k \rho L_t e_k(t+1) \\ -a_k(\gamma_k(t+1) - \rho) L_t e_k(t+1)$$

系统

$$x_k(t+1) = A(t)x_k(t) + B(t)u_k(t) + w_k(t+1)$$
$$y_k(t) = C(t)x_k(t) + v_k(t)$$

算法

$$u_{k+1}(t) = u_k(t) + a_k L_t e_k^*(t+1)$$

$$e_k^*(t) = \begin{cases} e_k(t), & \text{if } \gamma_k(t) = 1 \\ e_{k-1}^*(t), & \text{if } \gamma_k(t) = 0 \end{cases}$$

$$\frac{\tau_n^t}{n} \to 0 \quad \text{a.s.} \quad \forall t$$

$$u_{k+1}(t) = u_k(t) + a_k L_t e_{k-\tau_k^{t+1}}(t+1)$$

$$u_{k+1}(t) = u_k(t) + a_k L_t e_k(t+1)$$

$$\tau \sim G(\rho) \Rightarrow \mathbb{E}\tau = \frac{1}{\rho} \quad \mathbb{E}\tau^2 = \frac{2-\rho}{\rho^2}$$

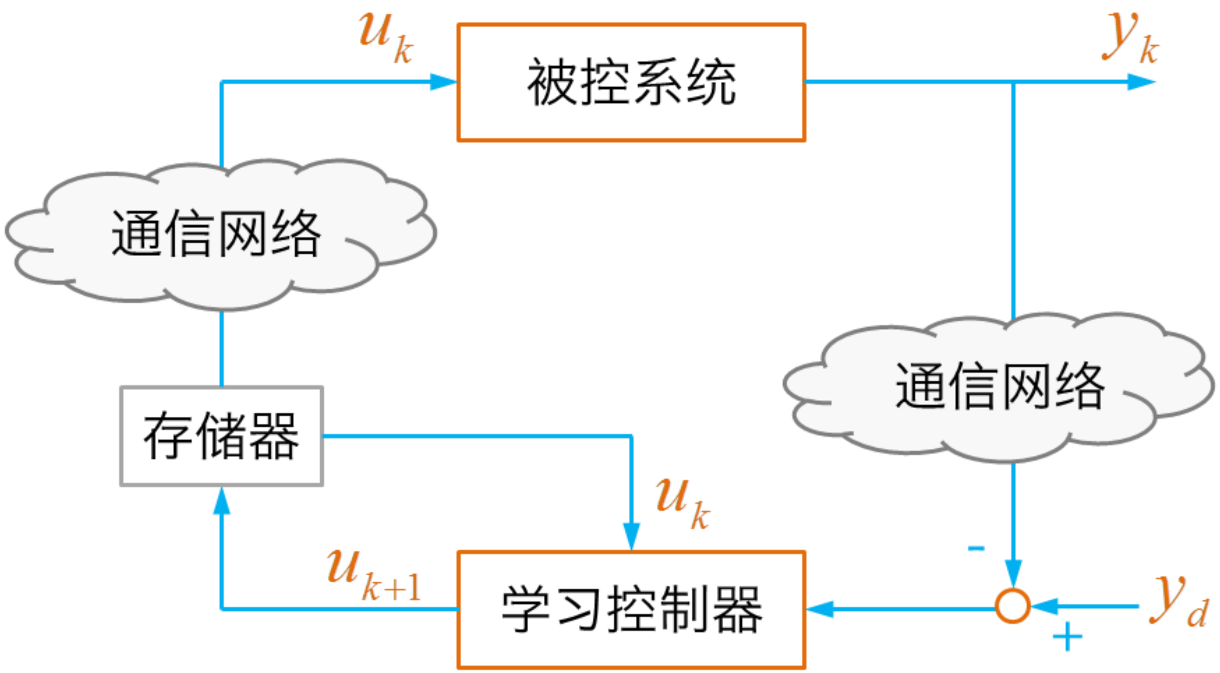
$$\sum_{n=1}^{\infty} \mathbb{P}\{\tau \ge n^{\frac{1}{2}}\} = \sum_{n=1}^{\infty} \mathbb{P}\{\tau^2 \ge n\} \le \mathbb{E}\tau^2 < \infty$$

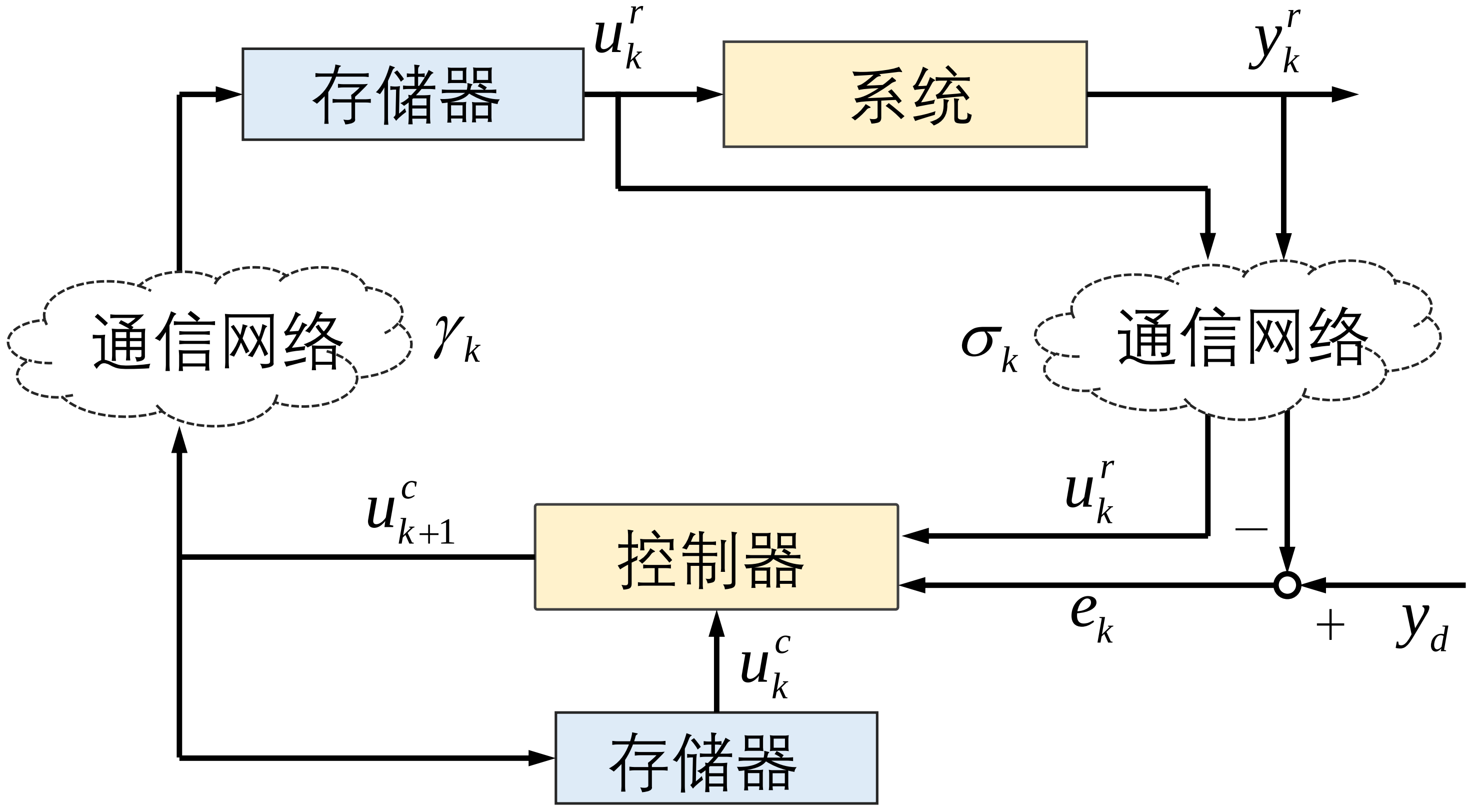
$$\Rightarrow \mathbb{P}\{\tau \ge n^{\frac{1}{2}} \quad \text{i.o.}\} = 0$$

$$a_k L_t (e_k(t+1) - e_{k-\tau_k^{t+1}}(t+1))$$

Dong Shen, Chao Zhang, Yun Xu. Information Sciences, vol. 381, pp. 352-370, 2017.

进一步，考虑在输入端存在随机数据丢包





存储器
u_k^r
系统
y_k^r
通信网络
γ_k
σ_k
通信网络
u_{k+1}^c
u_k^r
控制器
$-$
e_k
$+$
y_d
u_k^c
存储器

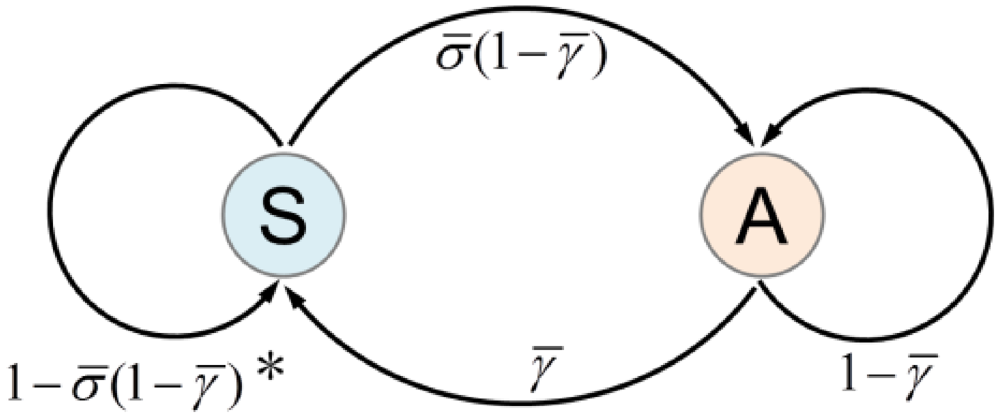
系统

$$x_k(t+1) = A_t x_k(t) + B_t u_k(t) + w_k(t)$$
$$y_k(t) = C_t x_k(t) + v_k(t)$$

算法

$$u^c_{k+1}(t) = \sigma_{k+1}(t) u^r_k(t) + (1-\sigma_{k+1}(t)) u^c_k(t) + \sigma_{k+1}(t) a_k L_t e_k(t+1)$$
$$u^r_{k+1}(t) = \gamma_{k+1}(t) u^c_{k+1}(t) + (1-\gamma_{k+1}(t)) u^r_k(t)$$

$$u^r_k(t) = u^c_k(t)$$



$$u^r_k(t) \neq u^c_k(t)$$

Dong Shen, Jian-Xin Xu. IEEE Transactions on Automatic Control, vol. 62, no. 11, pp. 5850-5857, 2017.

系统

$$x_k(t+1) = A_t x_k(t) + B_t u_k(t) + w_k(t)$$
$$y_k(t) = C_t x_k(t) + v_k(t)$$

算法

$$\begin{aligned} u_{k+1}^c(t) &= \sigma_{k+1}(t) u_k^r(t) + (1-\sigma_{k+1}(t)) u_k^c(t) \\ &\quad + \sigma_{k+1}(t) a_k L_t e_k(t+1) \\ u_{k+1}^r(t) &= \gamma_{k+1}(t) u_{k+1}^c(t) + (1-\gamma_{k+1}(t)) u_k^r(t) \end{aligned}$$

$$u_{k+1}^r(t) = u_k^r(t) + a_k \lambda_k(t) L_t e_k(t+1)$$

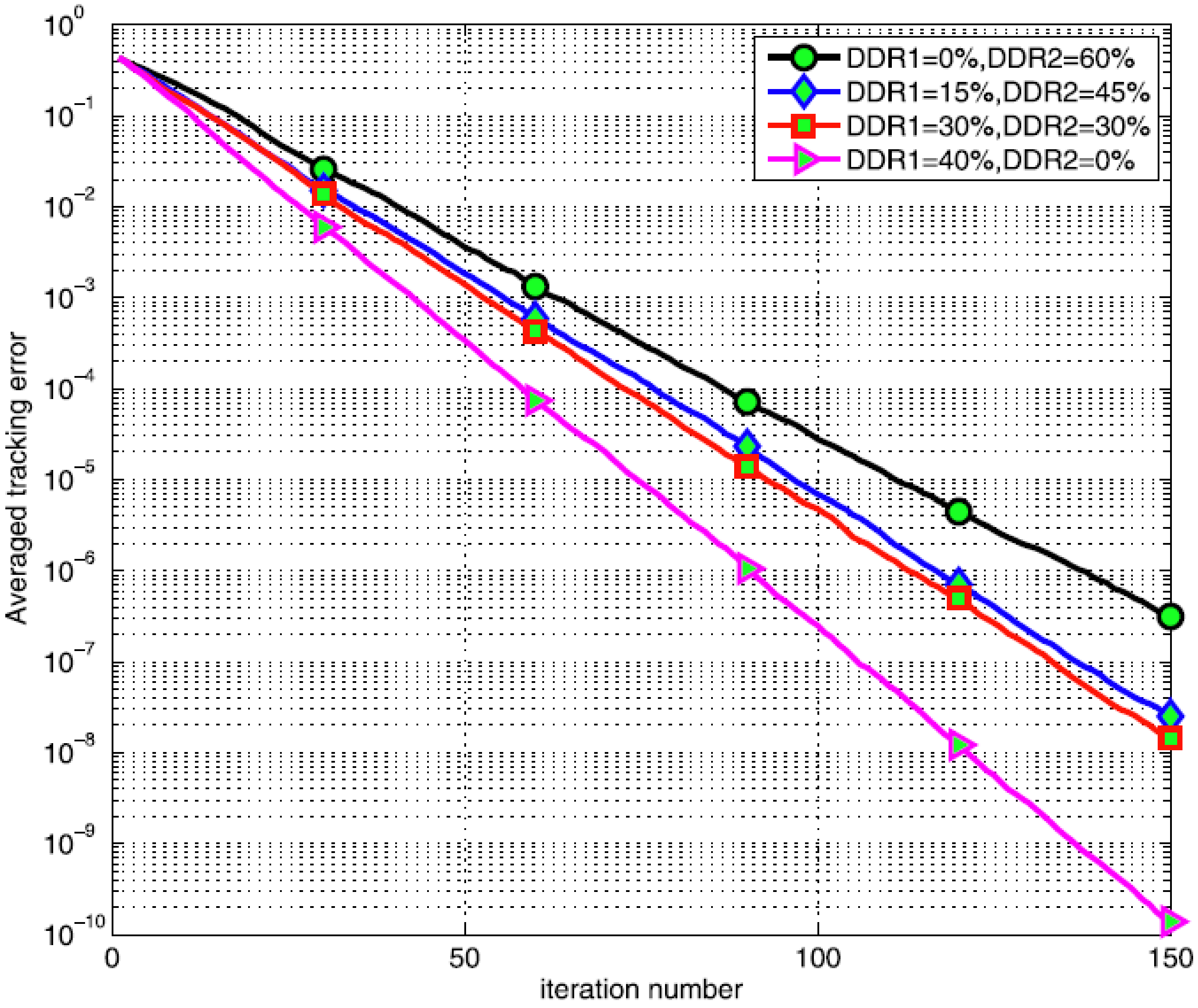
$$\|\Delta U_k^r\|_Q^2 \triangleq (\Delta U_k^r)^T Q \Delta U_k^r$$

$$\begin{aligned} U_{k+1}^r &= U_k^r + a_k \Lambda_k \mathcal{L} E_k \\ &= U_k^r + a_k \Lambda_k \mathcal{L}\mathcal{H} \Delta U_k^r - a_k \Lambda_k \epsilon_k \end{aligned}$$

$$\begin{aligned} \|\Delta U_{k+1}^r\|_Q^2 &= \|\Delta U_k^r\|_Q^2 + a_k^2 \|\Lambda_k \mathcal{L}\mathcal{H} \Delta U_k^r\|_Q^2 + a_k^2 \|\Lambda_k \epsilon_k\|_Q^2 \\ &\quad - a_k (\Delta U_k^r)^T ((\mathcal{L}\mathcal{H})^T \Lambda_k^T Q + Q \Lambda_k \mathcal{L}\mathcal{H}) \Delta U_k^r \\ &\quad + 2 a_k (\Delta U_k^r)^T Q \Lambda_k \epsilon_k \\ &\quad - 2 a_k (\Delta U_k^r)^T (\mathcal{L}\mathcal{H})^T \Lambda_k^T Q \Lambda_k \epsilon_k . \end{aligned}$$

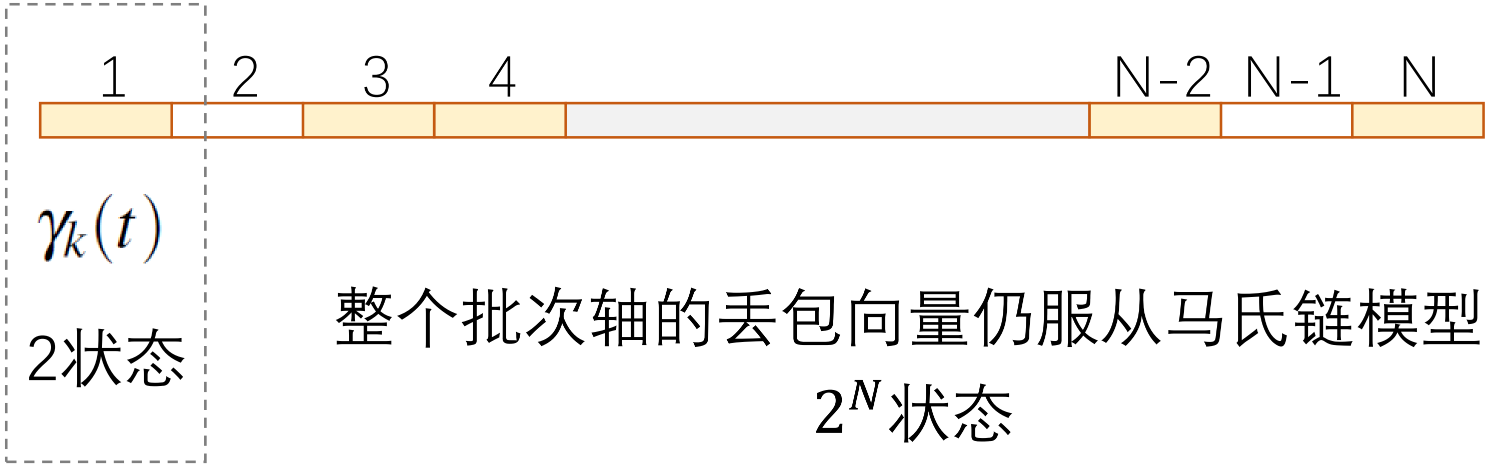
Dong Shen, Jian-Xin Xu. IEEE Transactions on Automatic Control, vol. 62, no. 11, pp. 5850-5857, 2017.

$$\mathbb{P}(renewal) = \frac{\overline{\gamma}}{\overline{\gamma} + \overline{\sigma} - \overline{\sigma} \cdot \overline{\gamma}} \cdot \overline{\sigma} \cdot \overline{\gamma} + \frac{\overline{\sigma} - \overline{\sigma} \cdot \overline{\gamma}}{\overline{\gamma} + \overline{\sigma} - \overline{\sigma} \cdot \overline{\gamma}} \cdot \overline{\gamma} = \frac{\overline{\sigma} \cdot \overline{\gamma}}{\overline{\gamma} + \overline{\sigma} - \overline{\sigma} \cdot \overline{\gamma}}.$$



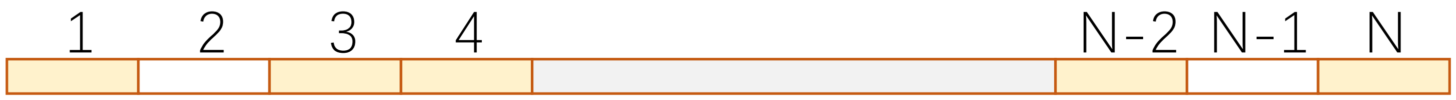
Dong Shen, Jian-Xin Xu. IEEE Transactions on Automatic Control, vol. 62, no. 11, pp. 5850-5857, 2017.

此时，再看马氏链模型就简单了



整个批次轴的丢包向量仍服从马氏链模型

2^N状态



$$\zeta_k \triangleq \{t \mid \gamma_k(t) = 1, 1 \leq t \leq N\}$$

$$\Xi \triangleq \{\zeta^{(1)}, \zeta^{(2)}, \cdots, \zeta^{(\iota)}\} \qquad \iota \triangleq 2^N$$

$$\zeta^{(1)} = \mathbf{Z}_1^N \qquad \zeta^{\iota} = \emptyset$$

时间独立性

$$\Xi_G \subseteq \Xi \qquad \bigcup_{i=1}^{K} \zeta_G^{(i)} = \zeta^{(1)}$$

时间可依赖

Dong Shen, Jian-Xin Xu, Xinghuo Yu. Submitted to Automatica, 2018.

Overview

背景介绍 → **数据丢包** → **相似问题拓展**

迭代学习控制

随机逼近

数据丢包模型

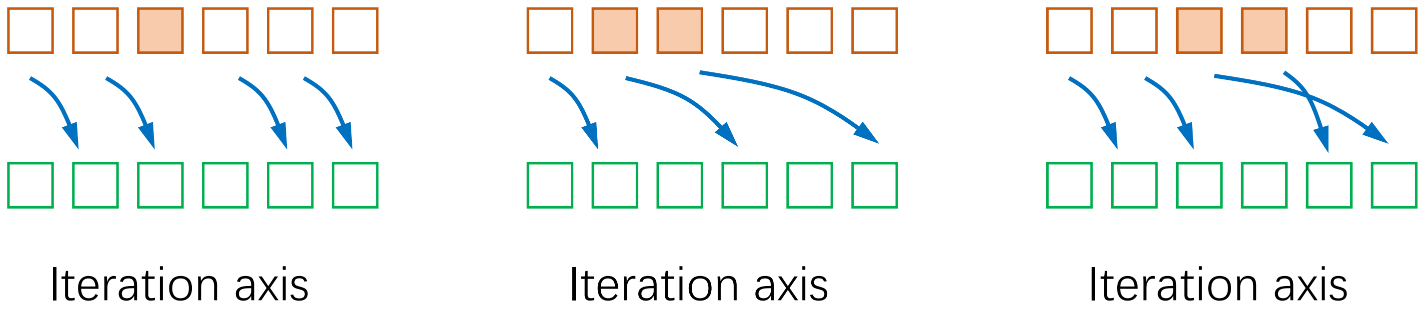
迭代学习控制问题

设计与分析

复杂通信问题

随机变长度问题

复杂通信环境下多种随机因素耦合



丢包　　延迟　　乱序

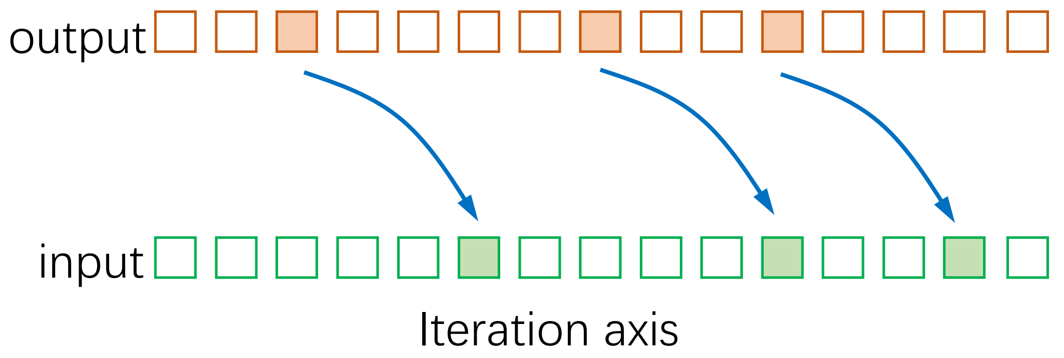
Dong Shen. IEEE Transactions on Neural Networks and Learning Systems. In press, 2018.

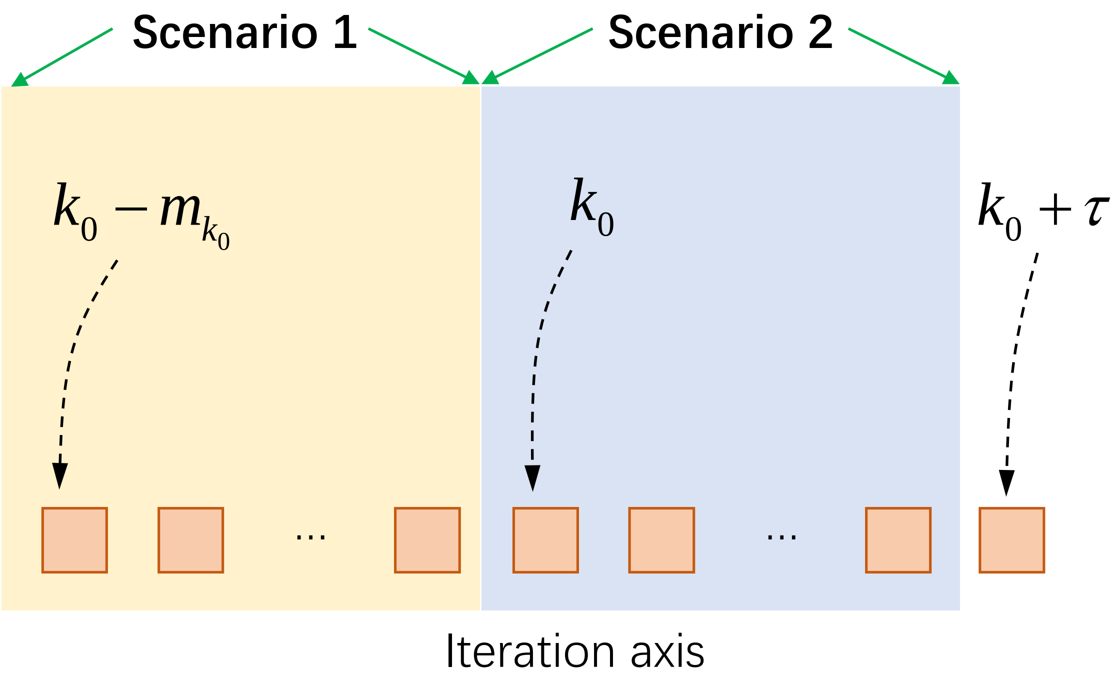
存储器数据更新机制

1. 数据包增加迭代戳
2. 接收到新数据包时
   若迭代戳新于当前存储数据包
      则存储新数据包
   若迭代戳未新于当前存储数据包
      则丢弃该数据包

控制器识别更新机制

当存储器接更新数据包时
若数据包迭代戳新于当前输入迭代戳
  则启动更新律，更新输入
若数据包迭代戳旧于当前输入迭代戳
  则保持当前输入不变





Dong Shen. IEEE Transactions on Neural Networks and Learning Systems. In press, 2018.

系统

$$x_k(t+1) = f(t, x_k(t)) + \mathbf{b}(t, x_k(t))u_k(t)$$
$$y_k(t) = c(t)x_k(t) + v_k(t)$$

随机序列模型

算法

$$u_{\tau_i}(t) = u_{\tau_{i-1}}(t) + a_{\tau_{i-1}} e_{\tau_i - n_{\tau_i}}(t+1)$$
$$u_k(t) = u_{\tau_i}(t)$$

间歇

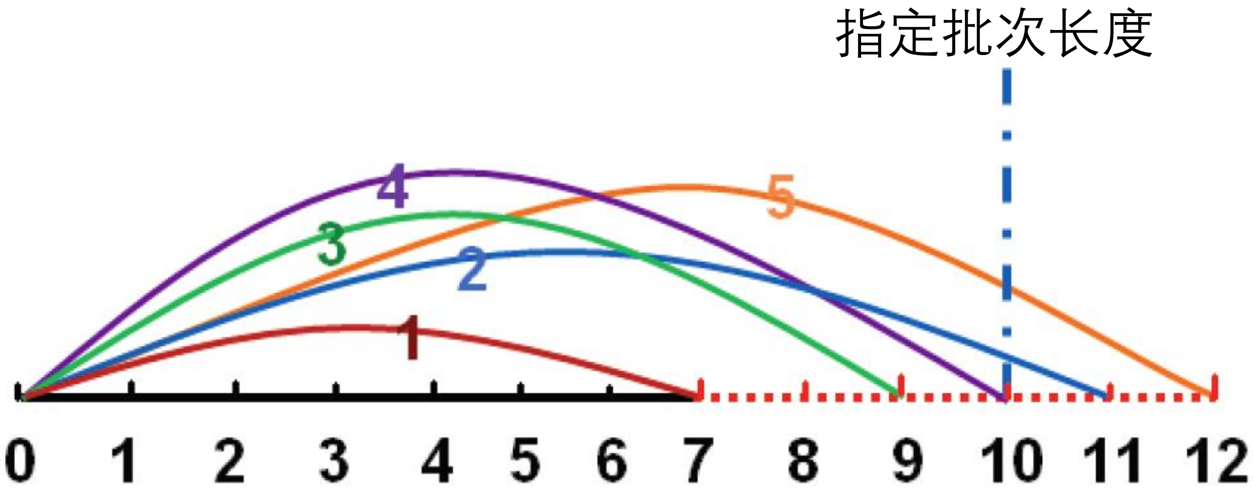
$$\tau_i < k \le \tau_{i+1} - 1$$

连续

$$u_{\tau_i}(t) = u_{\tau_i - 1}(t) + a_{\tau_i - 1} e_{\tau_i - n_{\tau_i}}(t+1)$$
$$u_k(t) = u_{k-1}(t) + a_{k-1} e_{\tau_i - n_{\tau_i}}(t+1)$$

Dong Shen. IEEE Transactions on Neural Networks and Learning Systems. In press, 2018.

批次长度随机变化问题



批次长度范围	$\{\overline{N}, \overline{N}+1, \ldots, N\}$
批次长度变量	$\overline{N} \leq N_k \leq N$
变量概率分布	$P(\mathcal{A}_{\overline{N}}) = p_1, P(\mathcal{A}_{\overline{N}+1}) = p_2, \ldots, P(\mathcal{A}_N) = p_m$
	$\mathcal{A}_l \quad \lfloor Y_k \rfloor_{N_k} = \lfloor Y_k \rfloor_l \qquad \lfloor \cdot \rfloor_{N_k} : \mathbb{R}^{Nq} \to \mathbb{R}^{N_k q}$
概率条件	$p_1 + p_2 + \cdots + p_m = 1$

↓

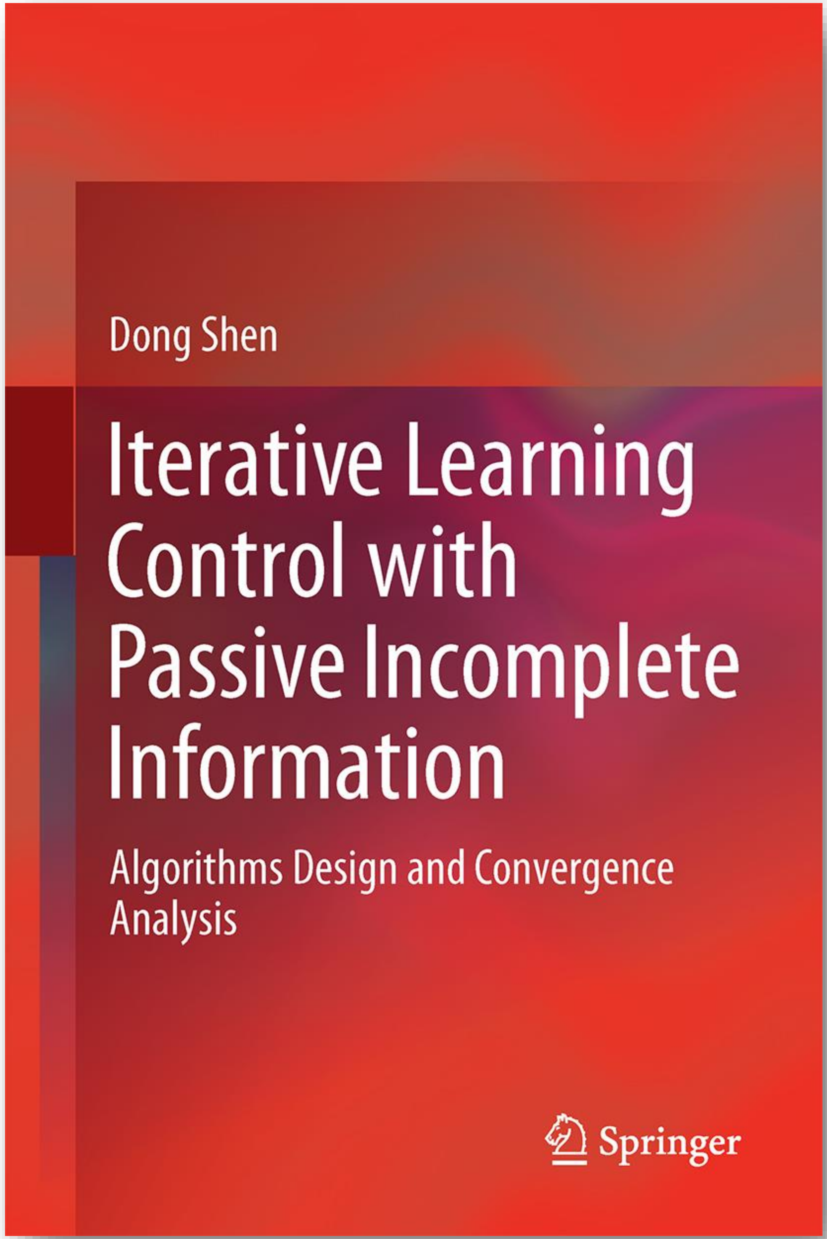
伯努利模型

Dong Shen, Wei Zhang, Youqing Wang, Chiang-Ju Chien. Automatica, vol. 63, no. 1, pp. 359-365, 2016.

Overview

背景介绍 → **数据丢包** → **相似问题拓展**

背景介绍	数据丢包	相似问题拓展
迭代学习控制	数据丢包模型	复杂通信问题
随机逼近	迭代学习控制问题	随机变长度问题
	设计与分析	



请您多批评指点!

所有论文均可在下述个人主页获取
http://shendongacademy.com/