AIを活用した
遠隔型の耐故障制御システムの構築

自然科学学域　中村幸紀

Cypher研究会（Society5.0研究支援プログラム成果報告）
オンライン開催　2021/7/30

はじめに

産業機器の故障：生産ラインの停止や事故の要因

▼

正確で速やかな故障検知が必要

はじめに

IoT技術の進展により通信機能を有するセンサや
制御装置が開発され，遠隔メンテナスへの応用が期待

▼

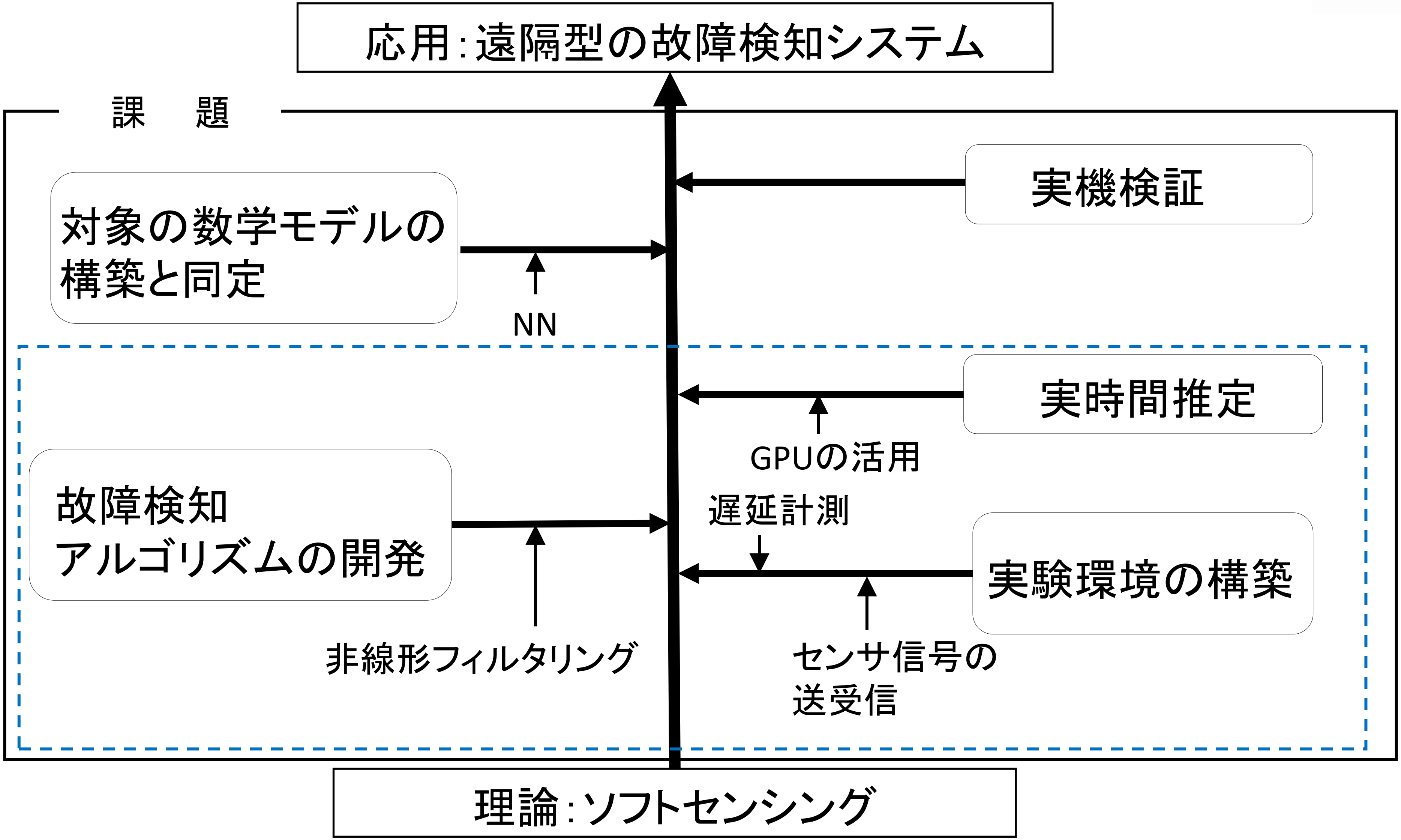
本研究：ネットワークを介した故障診断技術の開発

アプローチ

ソフトセンシング(状態推定)による故障検知

※ソフトセンシング：
・対象の数理モデル（e.g. 運動方程式，回路方程式）
・センサ出力
から対象の状態を推定すること

アプローチ

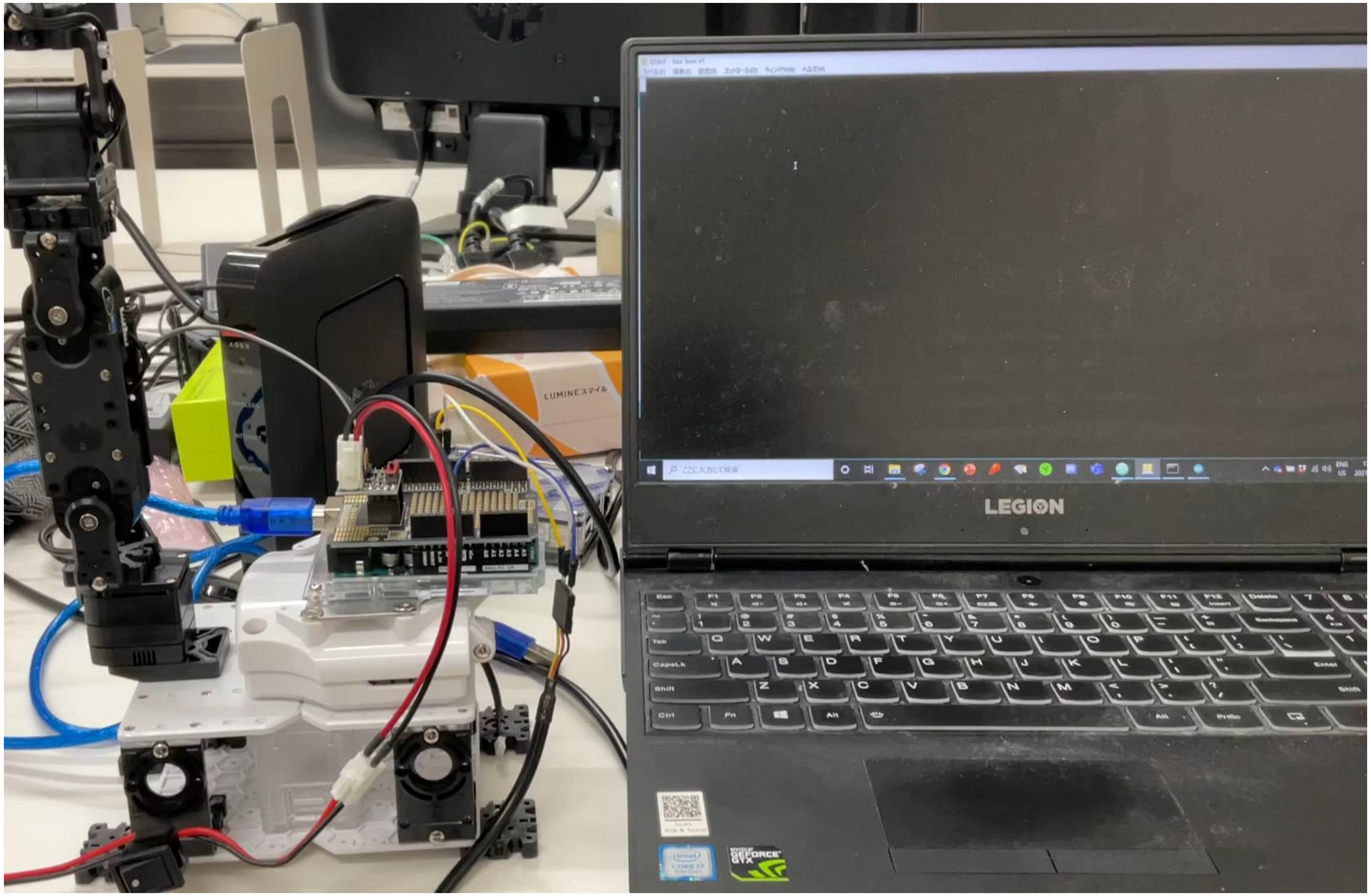


応用：遠隔型の故障検知システム
課　題
実機検証
対象の数学モデルの構築と同定
NN
実時間推定
GPUの活用
故障検知アルゴリズムの開発
遅延計測
実験環境の構築
非線形フィルタリング
センサ信号の送受信
理論：ソフトセンシング

1. 実験環境の構築

・無線を介してアームの関節角情報を送信
・故障検知アルゴリズムをMatlabにより実装

1. 実験環境の構築



センサ出力をPCに表示（無線通信による信号の送受信できることも確認）

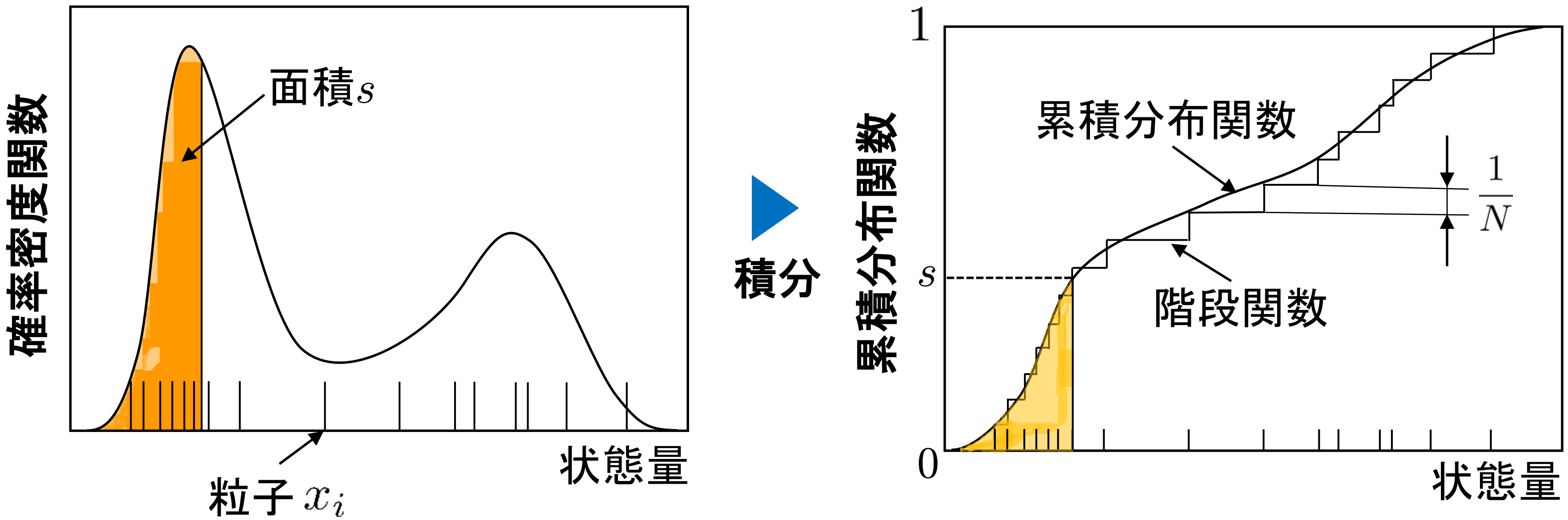
2. 故障検知アルゴリズムの開発

対象の数学モデルに基づくフィルタリング手法

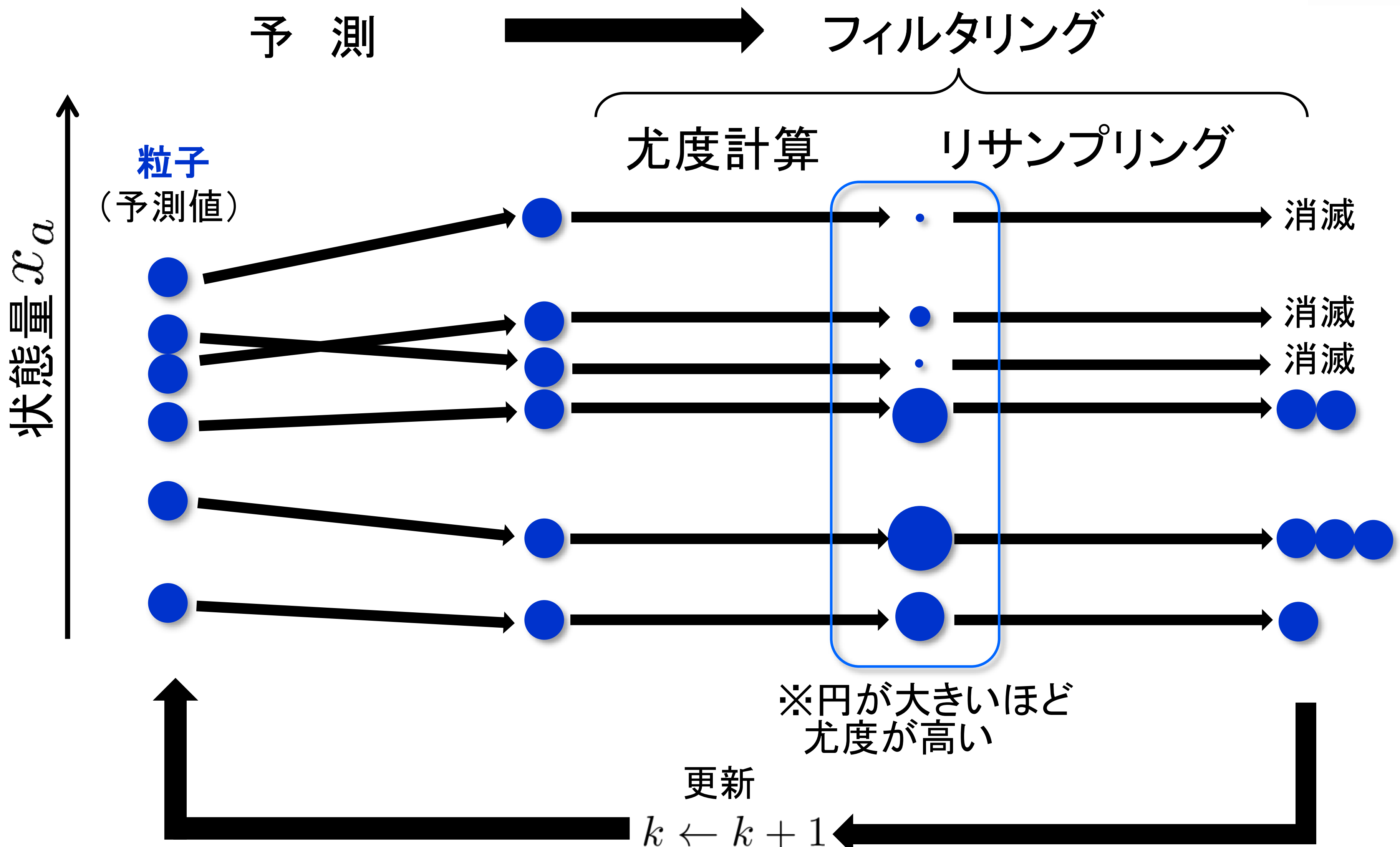
フィルタリング	数学モデル（状態方程式）	数学モデル（出力方程式）	雑　音
線形カルマンフィルタ	線　形	線　形	正規性
拡張カルマンフィルタ	非線形※ （※線形近似可）	非線形※ （※線形近似可）	正規性
アンサンブルカルマンフィルタ	非線形	線　形	システム雑音(任意) 観測雑音(正規性)
粒子フィルタ	非線形	非線形	任　意

粒子フィルタ

- 対象の状態量の確率分布を粒子で近似するアルゴリズム
- 非線形なシステム, 雑音の分布が非ガウスのモデルに対しても適用可能
- それぞれの粒子が状態量と尤度をもつ

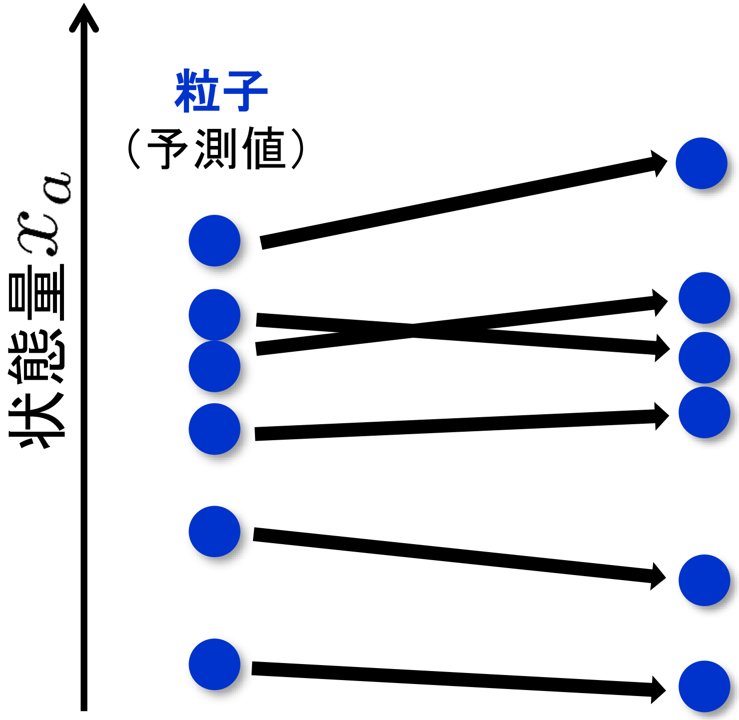


粒子フィルタのアルゴリズム



粒子フィルタのアルゴリズム(予測)

予　測



状態量x_aの時間変化を数学モデルより算出

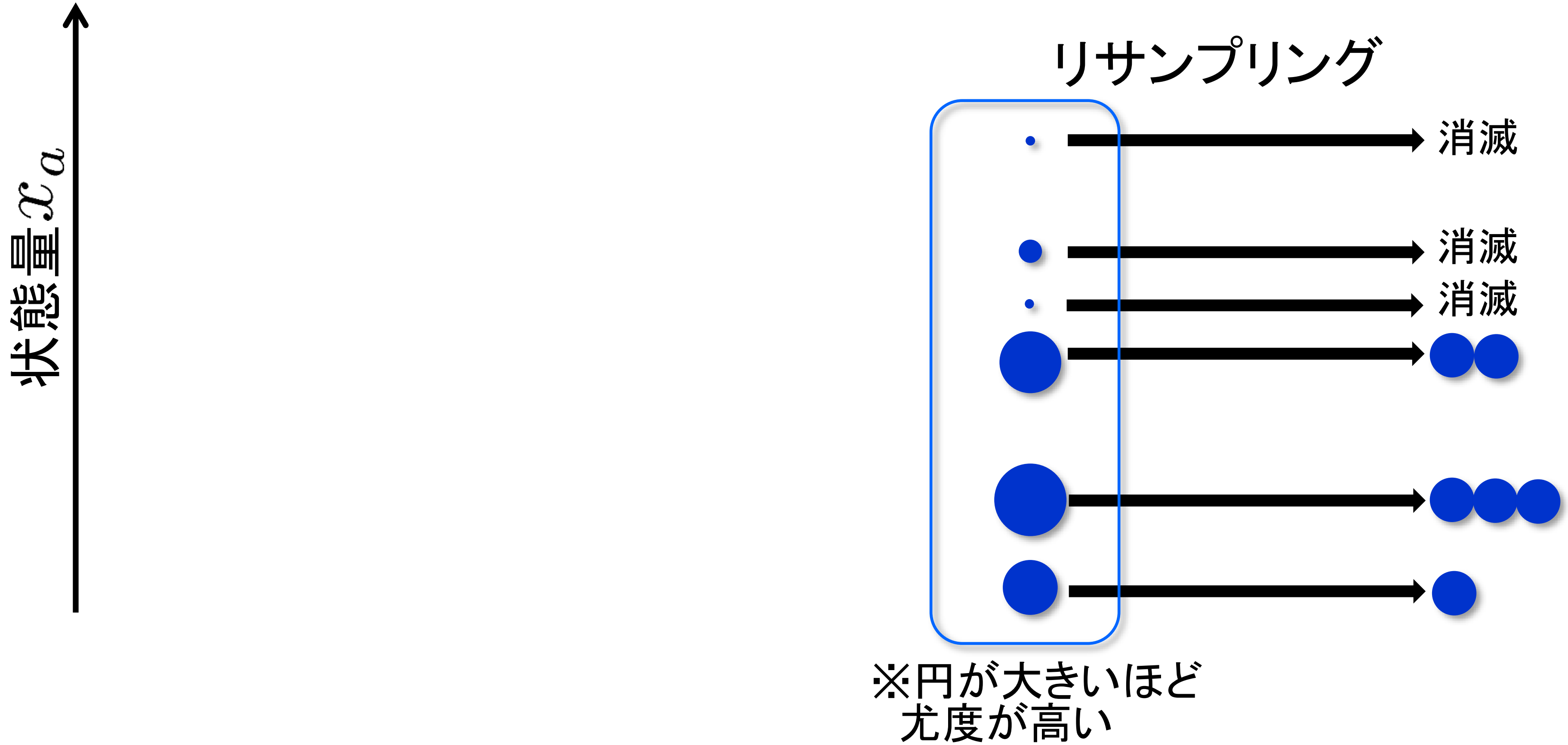
粒子フィルタのアルゴリズム(尤度計算)

尤度：センサ出力yが得られたときの状態量x_aの尤もらしさ.
条件付き確率$p(y|x_a)$

尤度計算

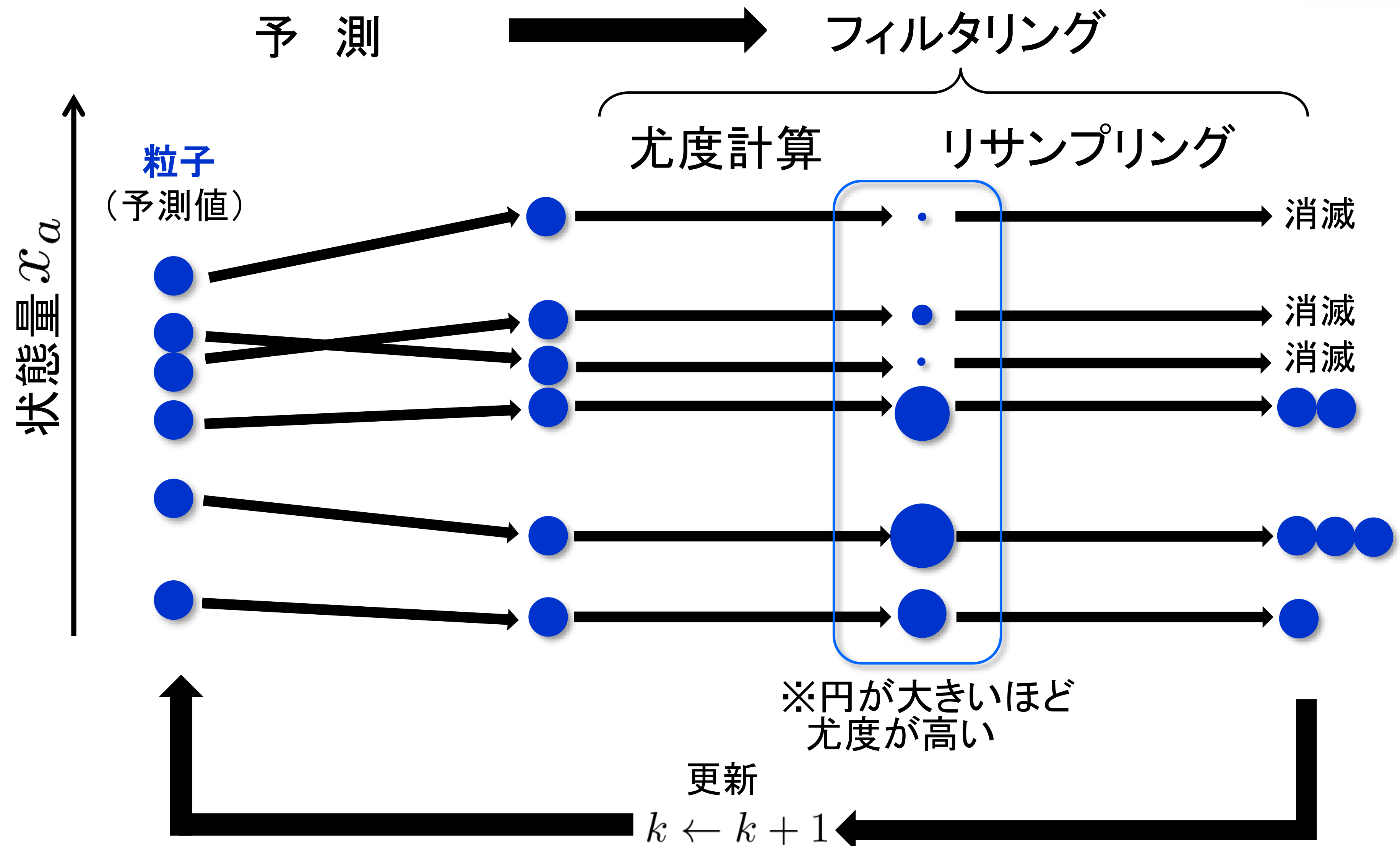
※円が大きいほど尤度が高い

粒子フィルタのアルゴリズム(リサンプリング)



尤度の高い粒子ほど複製を多く作成

粒子フィルタのアルゴリズム



2. 故障検知アルゴリズムの開発

課題：通信遅延が発生し，ネットワークの利用状況に応じて
遅延時間が不規則に変動
（故障検知の精度低下の要因）

▼

対策：タイムスタンプ(信号の送信時刻情報)に基づく推定法※

▼

本研究：故障検知へ応用したアルゴリズムの開発

※中村ら，電気学会論文誌C, 2019

故障検知のアルゴリズム

対象の数学モデル: 離散時間非線形システム

$$x(k) = f(x(k-1), \theta, u(k), v(k)) \quad k = 1, 2, \cdots$$

$$y(k) = h(x(k), \xi) + w(k)$$

- 状態量: $x \in \mathbb{R}^n$
- パラメータ: $\theta \in \mathbb{R}^{m_1}, \ \xi \in \mathbb{R}^{m_2}$
- 入力: $u \in \mathbb{R}^l$
- システム雑音: $v \in \mathbb{R}^n$(確率密度関数$\tilde{v}$は既知)
- 出力: $y \in \mathbb{R}^p$
- 観測雑音: $w \in \mathbb{R}^p$(確率密度関数$\tilde{w}$は既知)
- 遅延: d（遅延の最大値h_dは既知）

仮定: 遅延dは時変かつ有界($h_d = \max_{k \in \mathbb{N}} d(k) < \infty$)

故障検知のアルゴリズム

対象の数学モデル: 離散時間非線形システム

$$x(k) = f(x(k-1), \theta, u(k), v(k)) \quad k = 1, 2, \cdots$$

$$y(k) = h(x(k), \xi) + w(k)$$

- 状態量: $x \in \mathbb{R}^n$
- パラメータ: $\theta \in \mathbb{R}^{m_1}$, $\xi \in \mathbb{R}^{m_2}$
- 入力: $u \in \mathbb{R}^l$
- システム雑音: $v \in \mathbb{R}^n$(確率密度関数$\tilde{v}$は既知)
- 出力: $y \in \mathbb{R}^p$
- 観測雑音: $w \in \mathbb{R}^p$(確率密度関数$\tilde{w}$は既知)
- 遅延: d（遅延の最大値h_dは既知）

仮定: 遅延dは時変かつ有界($h_d = \max_{k \in \mathbb{N}} d(k) < \infty$)

出力yからセンサの故障検出を行う

OKAYAMA UNIV.

故障検知のアルゴリズム

OKAYAMA UNIV.

対象の数学モデル

$$x(k) = f(x(k-1), \theta, u(k), v(k)) \quad k = 1, 2, \cdots$$

$$y(k) = h(x(k), \xi) + w(k)$$

▼

センサの故障検知

$$y(k) = \mathrm{diag}(\alpha_1, \alpha_2, \cdots, \alpha_p) h(x(k), \xi) + w(k)$$

$\alpha_1, \alpha_2, \cdots, \alpha_p$:センサ故障の有無を判別するパラメータ（検知パラメータ）

$$\tilde{h}(x(k), \tilde{\xi}) = \mathrm{diag}(\alpha_1, \alpha_2, \cdots, \alpha_p) h(x(k), \xi)$$

$$y(k) = \tilde{h}(x(k), \tilde{\xi}), \quad \tilde{\xi} = \begin{bmatrix} \xi & \alpha_1 & \alpha_2 & \cdots & \alpha_p \end{bmatrix}^{\mathrm{T}}$$

$\tilde{\xi}$ を推定することでセンサの故障を検知

故障検知のアルゴリズム

検知パラメータ:センサ故障の有無を表すパラメータ

$$y(k) = \mathrm{diag}(\alpha_1, \alpha_2, \cdots, \alpha_p) h(x(k), \xi\ ) + w(k)$$

センサ正常時 $\alpha_i = 1$

センサ故障時 $\alpha_i = 1$　とは限らない

例: $y(k) = \alpha_1 x(k) + w(k)$　　$y, x \in \mathbb{R}$ (スカラ)

故障: $y(k) = y(k_f),\ \ k \geq k_f$　　k_f:故障時間

$y(k_f) = \alpha_1 x(k) + w(k),\ \ k \geq k_f$

この場合 $\alpha_1 = 1$ とは限らない

本手法:検知パラメータ α_i の値により故障の有無を判断

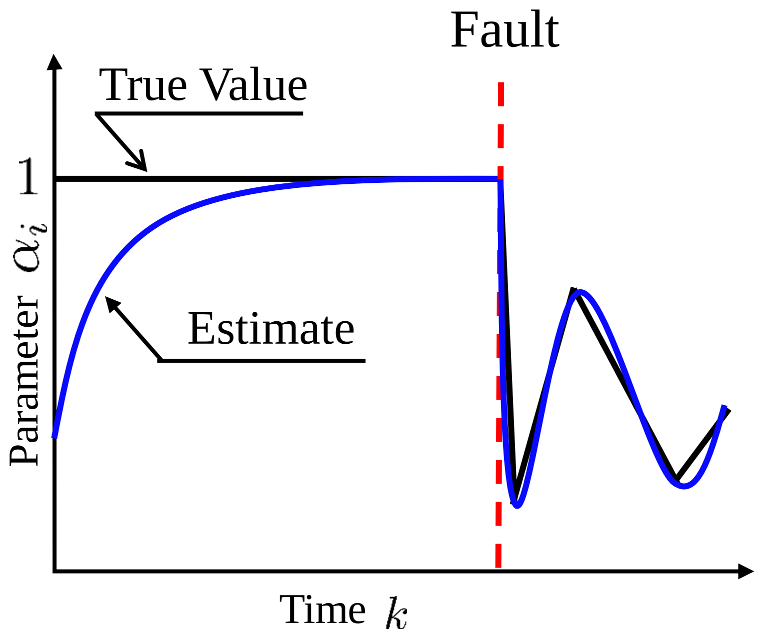
故障検知のアルゴリズム

$$x(k) = f(x(k-1), \theta\ , u(k), v(k))$$

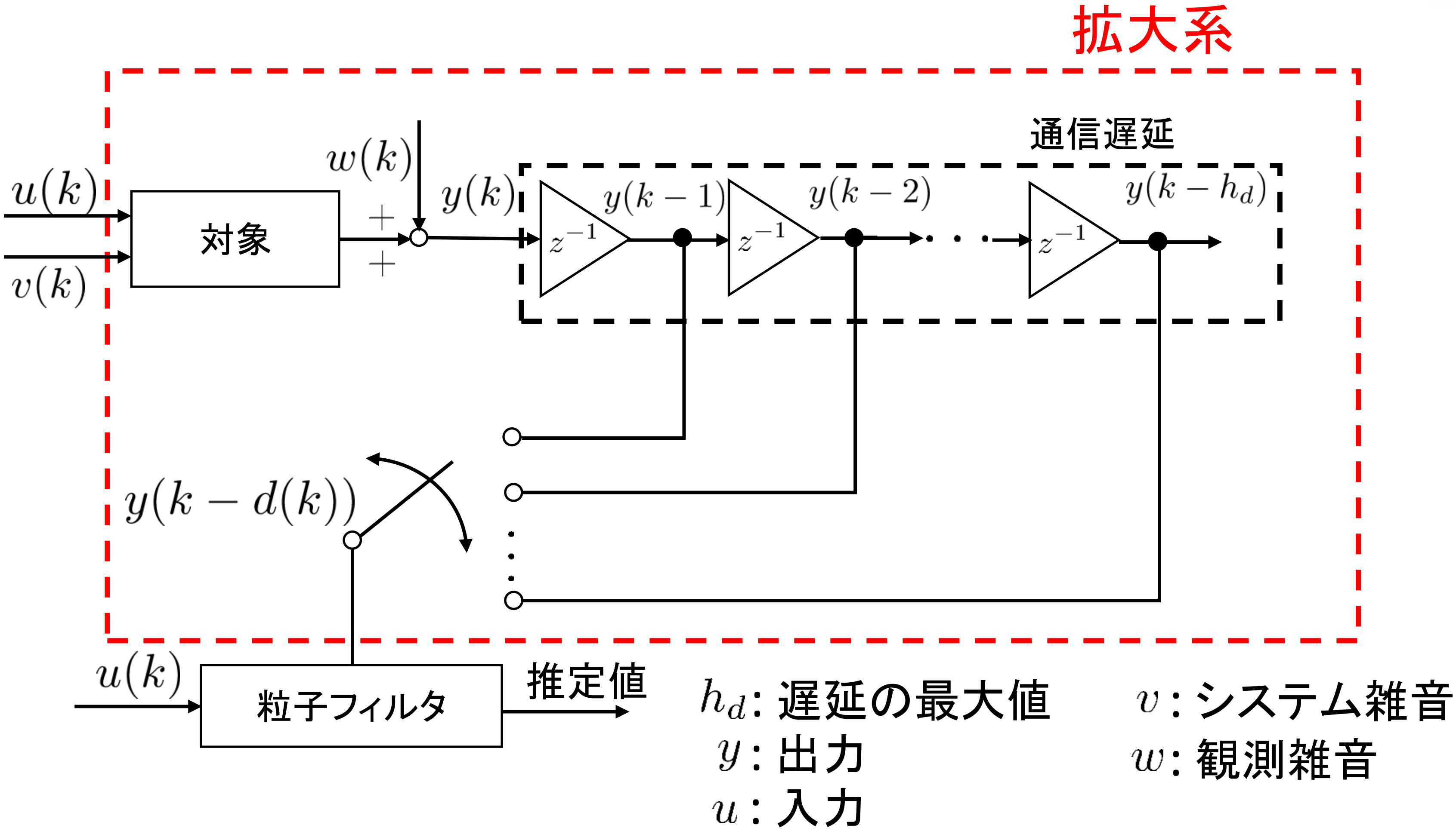
$$y(k) = \mathrm{diag}(\alpha_1, \alpha_2, \cdots, \alpha_p) h(x(k), \xi\ ) + w(k)$$

故障検出に用いる
パラメータ

- センサ正常時検出パラメータ α_i は1
- センサ故障によりパラメータ α_i が変化
- 検出パラメータ α_i の変化を
粒子フィルタにより推定し，
推定値に基づいて故障を検知



推定対象の記述



拡大系に対して推定を行う

拡大系の定式化

- 拡大系の状態ベクトルx_a

$$x_a(k) = \begin{bmatrix} x^{\mathrm{T}} & \theta^{\mathrm{T}} & \xi^{\mathrm{T}} & y^{\mathrm{T}}(k-1) & \cdots & y^{\mathrm{T}}(k-h_d) & \alpha \end{bmatrix}^{\mathrm{T}}$$

$$\alpha = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_p \end{bmatrix}^{\mathrm{T}}$$ 検知パラメータ

- 拡大系の数学モデル

$$x_a(k) = f_a(x_a(k-1), u(k), v_a(k))$$

$$v_a(k) = \begin{bmatrix} v^{\mathrm{T}}(k) & w^{\mathrm{T}}(k) \end{bmatrix}^{\mathrm{T}}$$

$$y(k-d(k)) = C_a(d(k))x_a(k) + q_a(k)$$

q_a:通信時に加わる雑音

拡大系の数学モデル

$$f_a(x_a(k-1),u(k),v_a(k)) = \begin{bmatrix} f(x(k-1),\theta,u(k),v(k)) \\ \theta \\ \xi \\ \mathrm{diag}(\alpha_1,\alpha_2,\cdots,\alpha_p)h(x(k),\xi)+w(k) \\ y_d(k) \\ \alpha \end{bmatrix}$$

$$y_d(k) = \begin{bmatrix} y^{\mathrm{T}}(k-1) & y^{\mathrm{T}}(k-2) & \cdots & y^{\mathrm{T}}(k-h_d+1) \end{bmatrix}^{\mathrm{T}}$$

$$C_a(d) = \begin{bmatrix} \mathbf{0}_{p\times r_1} & I_p & \mathbf{0}_{p\times r_2} \end{bmatrix}$$

$$r_1 = n+m_1+m_2+pd-p \qquad r_2 = p(h_d-d)$$

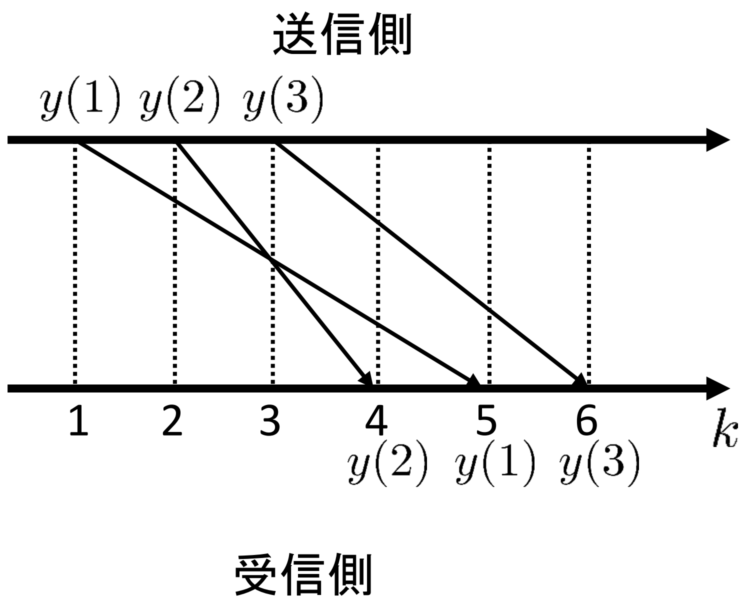
$$\boxed{y(k-d(k)) = C_a(d(k))x_a(k)+q_a(k)}$$

C_a が遅延 d に応じて変化するため未知

▶ 数学モデルを用いる粒子フィルタの適用が困難

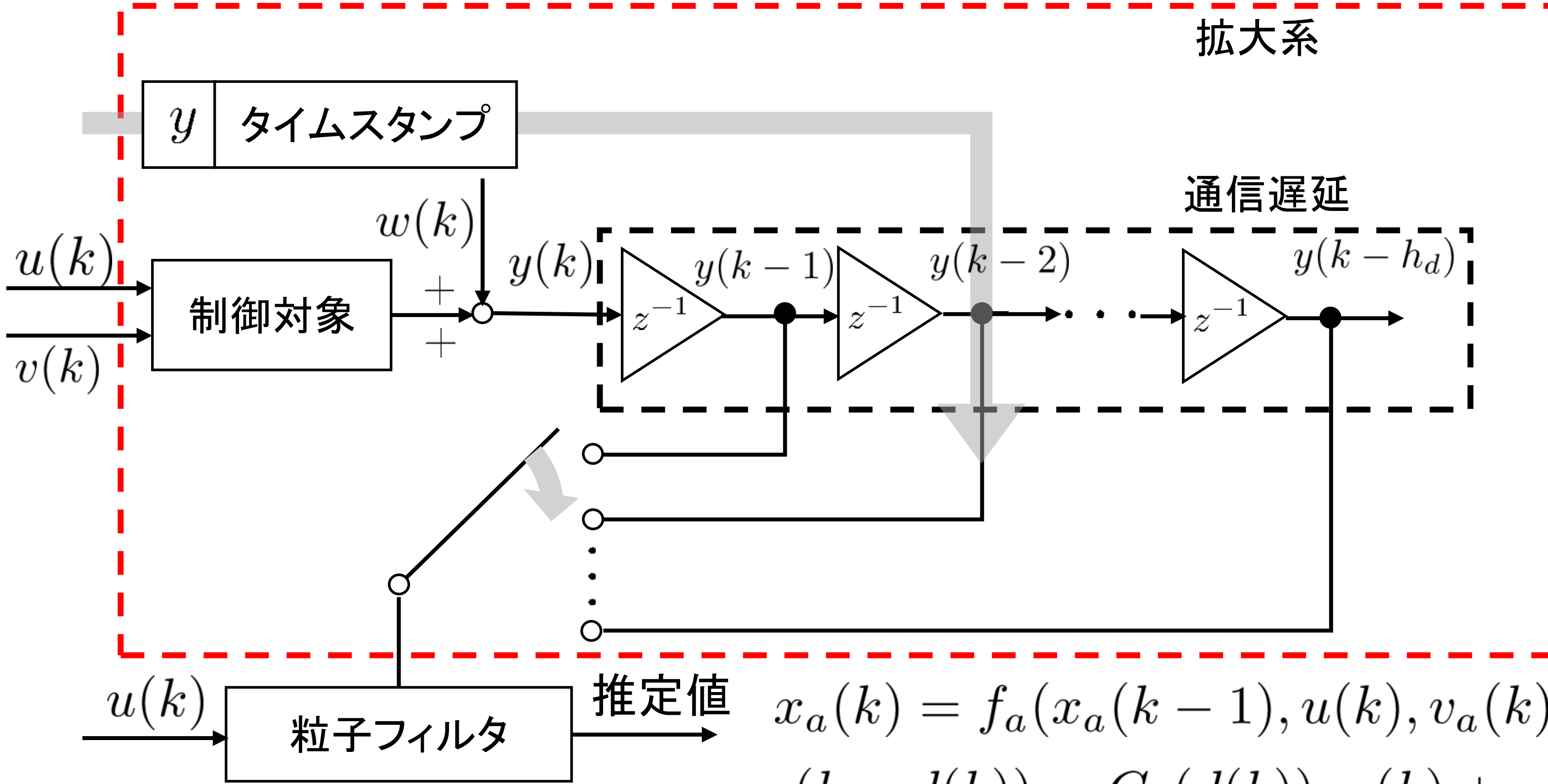
タイムスタンプ(送信時刻情報)

- 送信側は出力の送信時にタイムスタンプを付加して送信
- 受信側は出力の受信時に現在時刻とタイムスタンプの差から遅延dを計測



受信時刻	$k=4$	$k=5$	$k=6$
出力	$y(2)$	$y(1)$	$y(3)$
遅延	$d=2$	$d=4$	$d=3$

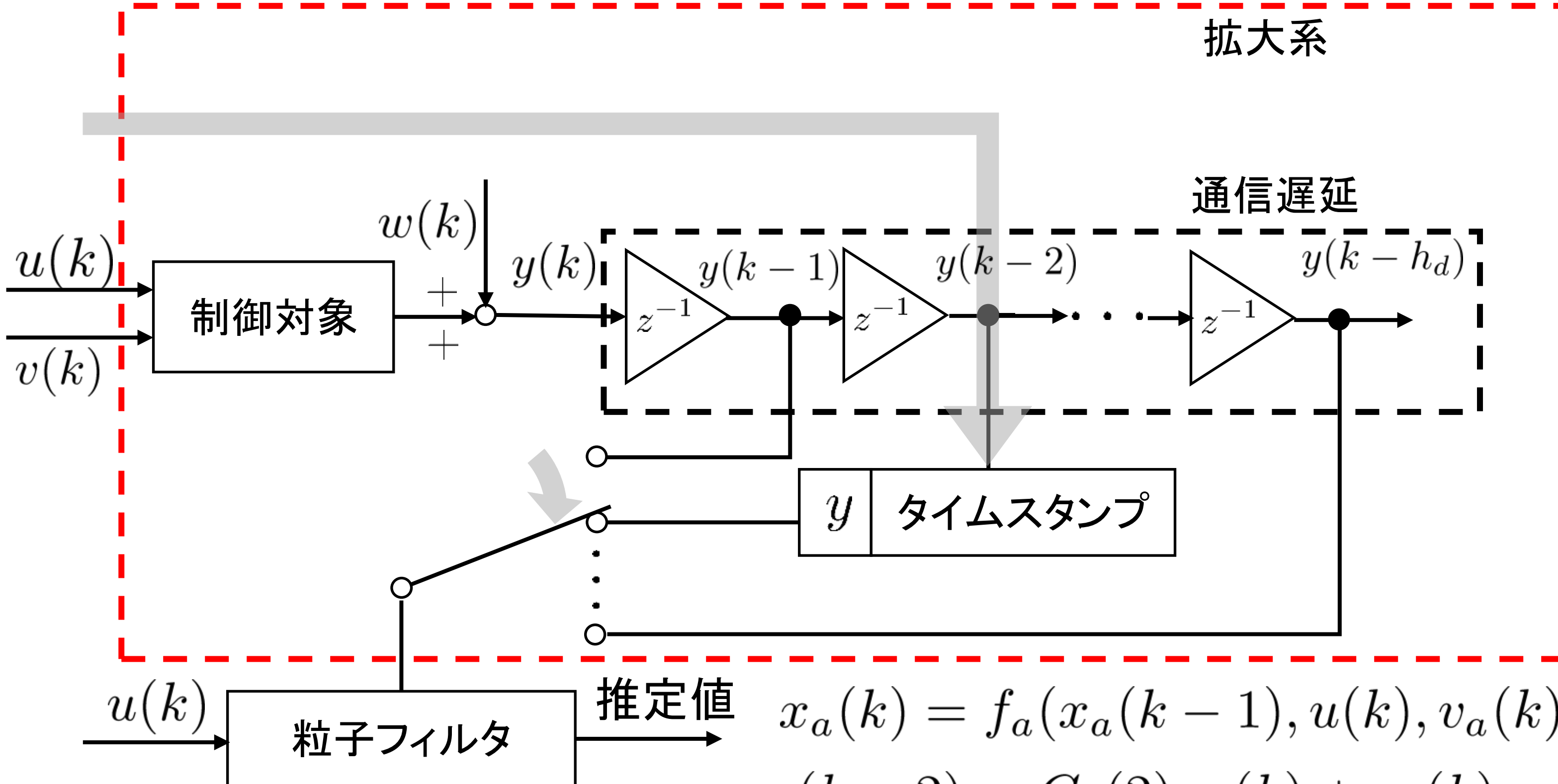
タイムスタンプ



$$x_a(k) = f_a(x_a(k-1), u(k), v_a(k))$$

$$y(k-d(k)) = \underbrace{C_a(d(k))}_{\text{未知}} x_a(k) + q_a(k)$$

タイムスタンプ

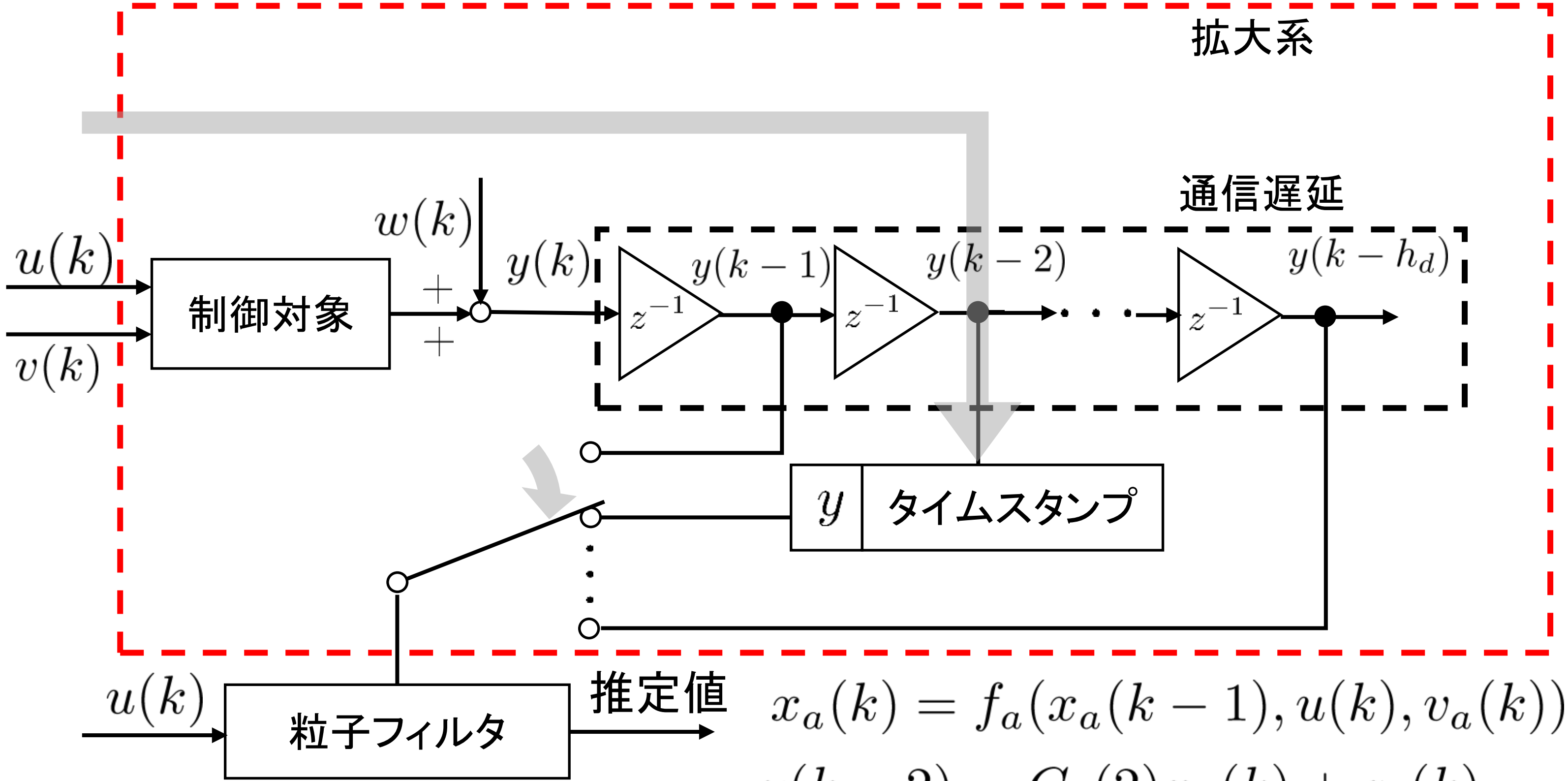


$$x_a(k) = f_a(x_a(k-1), u(k), v_a(k))$$

$$y(k-2) = \underbrace{C_a(2)}_{\text{既知}} x_a(k) + q_a(k)$$

タイムスタンプ

OKAYAMA UNIV.



$$x_a(k) = f_a(x_a(k-1), u(k), v_a(k))$$

$$y(k-2) = \underbrace{C_a(2)}_{\text{既知}} x_a(k) + q_a(k)$$

タイムスタンプを用いることでC_aを既知とみなすことが可能

▶ 粒子フィルタを適用することが可能

シミュレーションによる検証

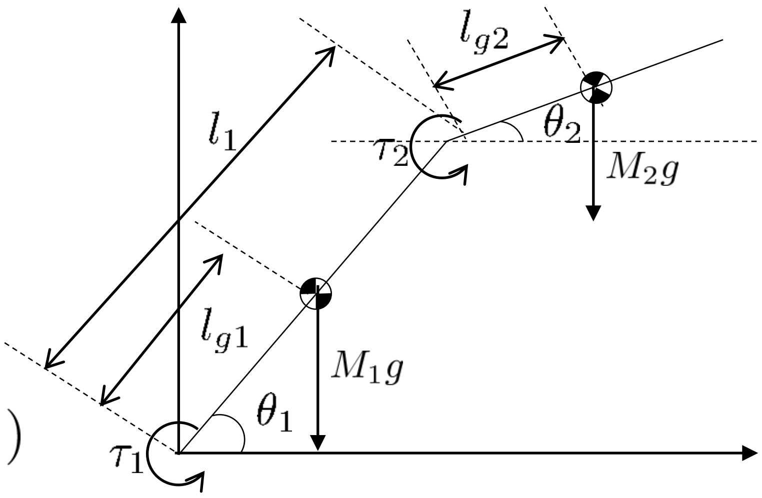
OKAYAMA UNIV.

- 対象 : 2リンクアーム
- 目的 : 角度センサの故障検知
- 故障の種類:出力が一定となる固着型故障

シミュレーション条件

- 2リンクアームの数学モデル

$$\dot{x}(t) = \begin{bmatrix} x_3(t) \\ x_4(t) \\ \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix}^{-1} \begin{bmatrix} \tau_1 - c_1 - d_1 \\ \tau_2 - c_2 - d_2 \end{bmatrix} \end{bmatrix}$$



$$a_1 = I_1 + I_2 + M1l_{g2}^2 + M_2(l_1^2 + 2l_1l_{g2}\cos\theta_2)$$

$$b_1 = I_2 + M_2(l_{g2}^2 + l_1l_{g2}\cos\theta_2)$$

$$c_1 = M_1gl_{g1}\cos\theta_1 - M_2l_1l_{g2}\sin\theta_2(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) + M_2g\{l_1\cos\theta_1 + l_{g2}\cos(\theta_1+\theta_2)\}$$

$$a_2 = I_2 + M_2(l_{g2}^2 + l_1l_{g2}\cos\theta_2)$$

$$b_2 = M_2l_{g2}^2 + I_2$$

$$c_2 = M_2l_1l_{g2}\dot{\theta}_1^2\sin\theta_2 + M_2gl_{g2}\cos(\theta_1+\theta_2)$$

B_1, B_2 :粘性摩擦係数

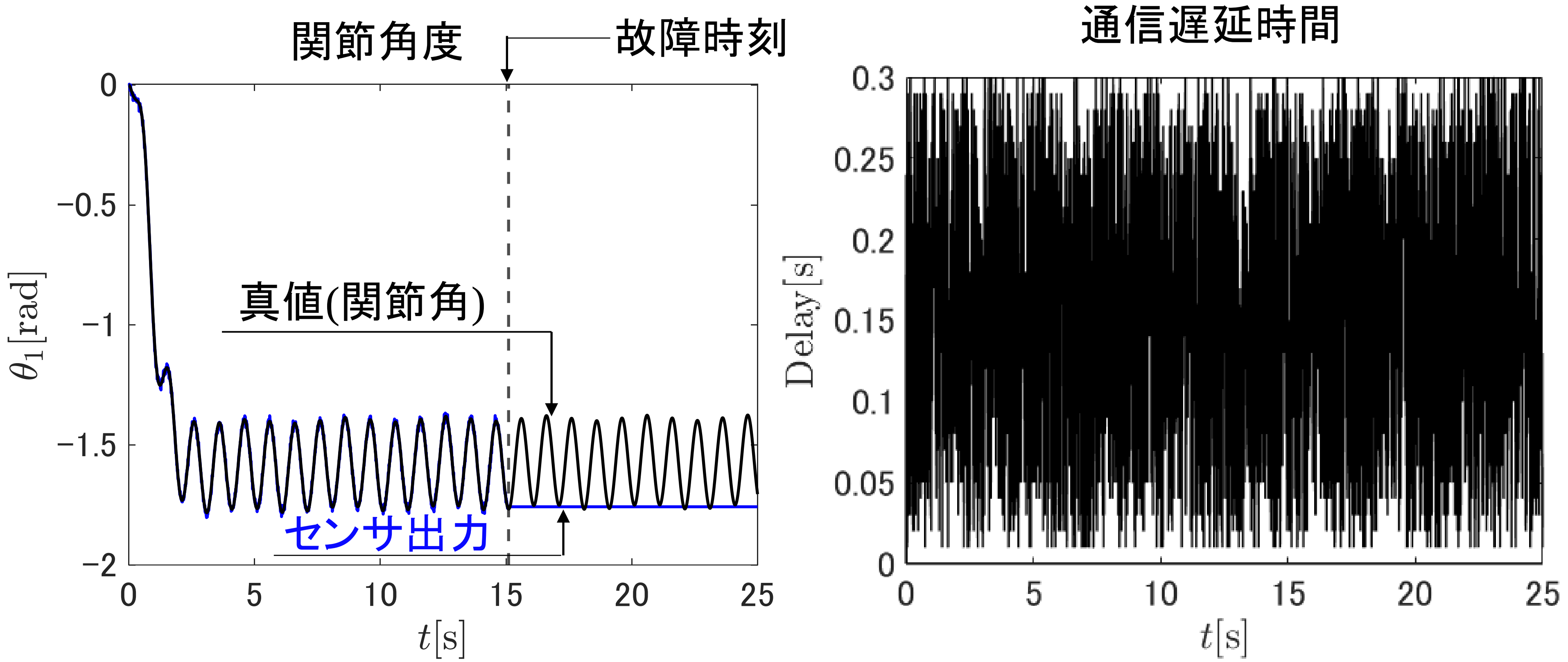
I_1, I_2 :慣性モーメント

$$d_1 = B_1\dot{\theta}_1$$

$$d_2 = B_2\dot{\theta}_2$$

シミュレーション条件

- 開始から約15秒以降，センサ故障により出力 y が固定
- 遅延 d は一様分布に従うとして生成



評価方法

検知パラメータの推定値を評価

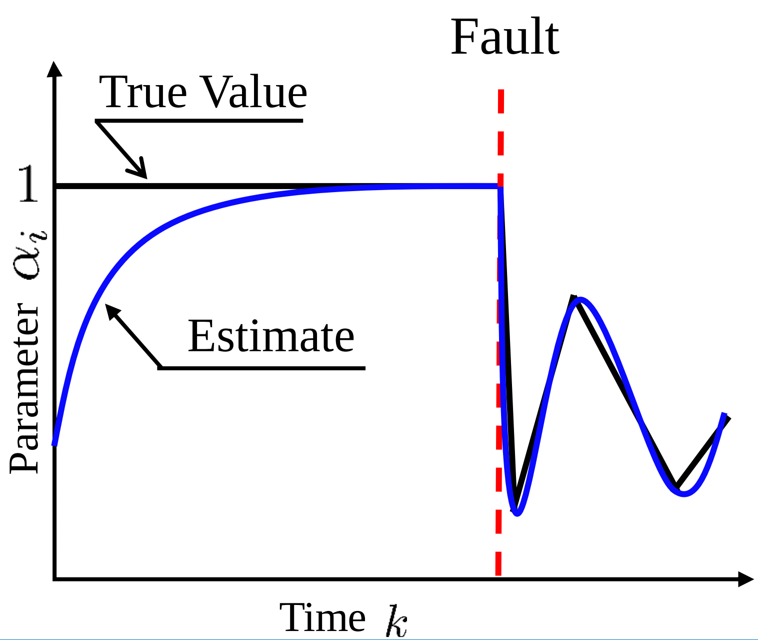
対象の数学モデル

$$x(k) = f(x(k-1), \theta\ , u(k), v(k))$$

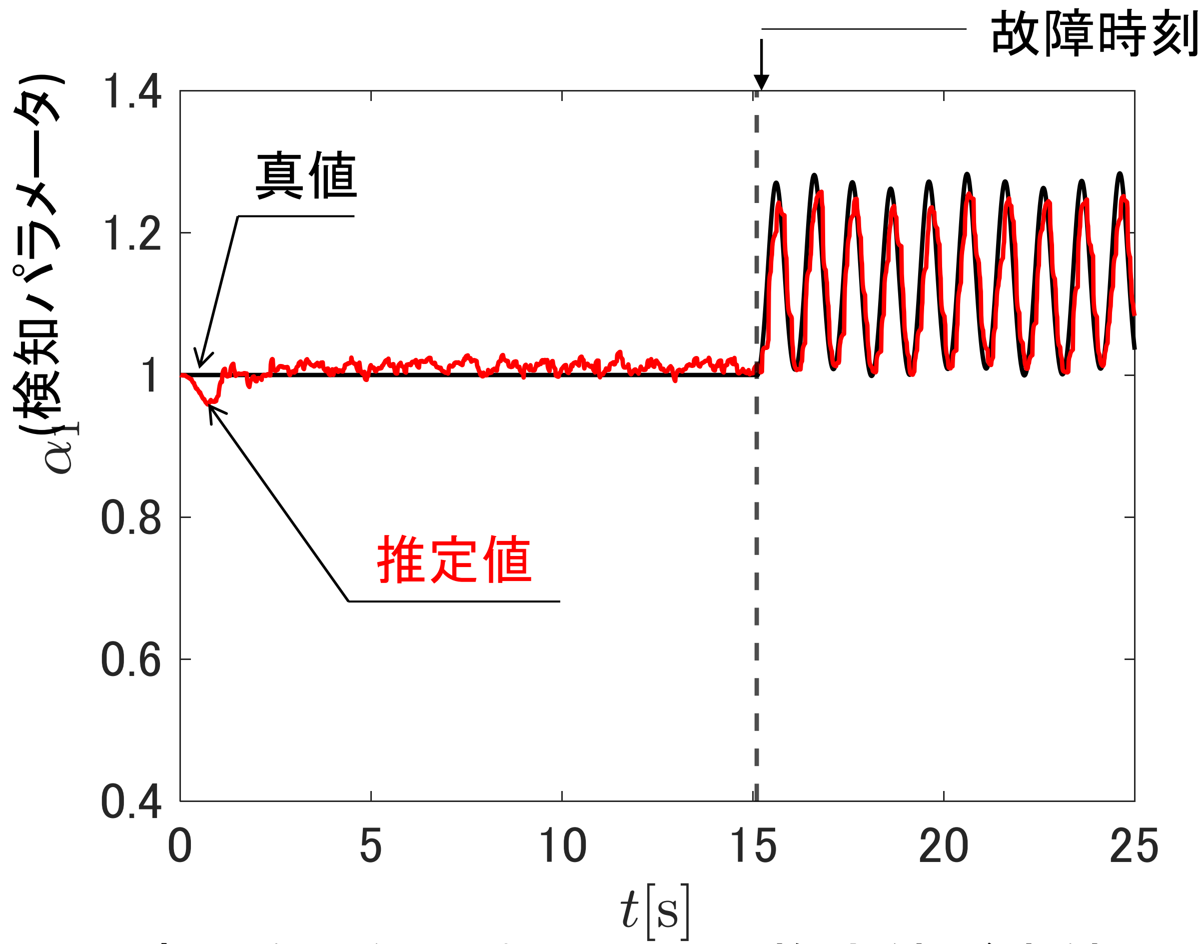
$$y(k) = \mathrm{diag}(\alpha_1, \alpha_2, \cdots, \alpha_p) h(x(k), \xi\ ) + w(k)$$

検知パラメータ

- センサ正常時検知パラメータ α_i は1
- センサ故障によりパラメータα_iが変化
- 検知パラメータα_iの変化を状態推定により把握し，故障検出を行う



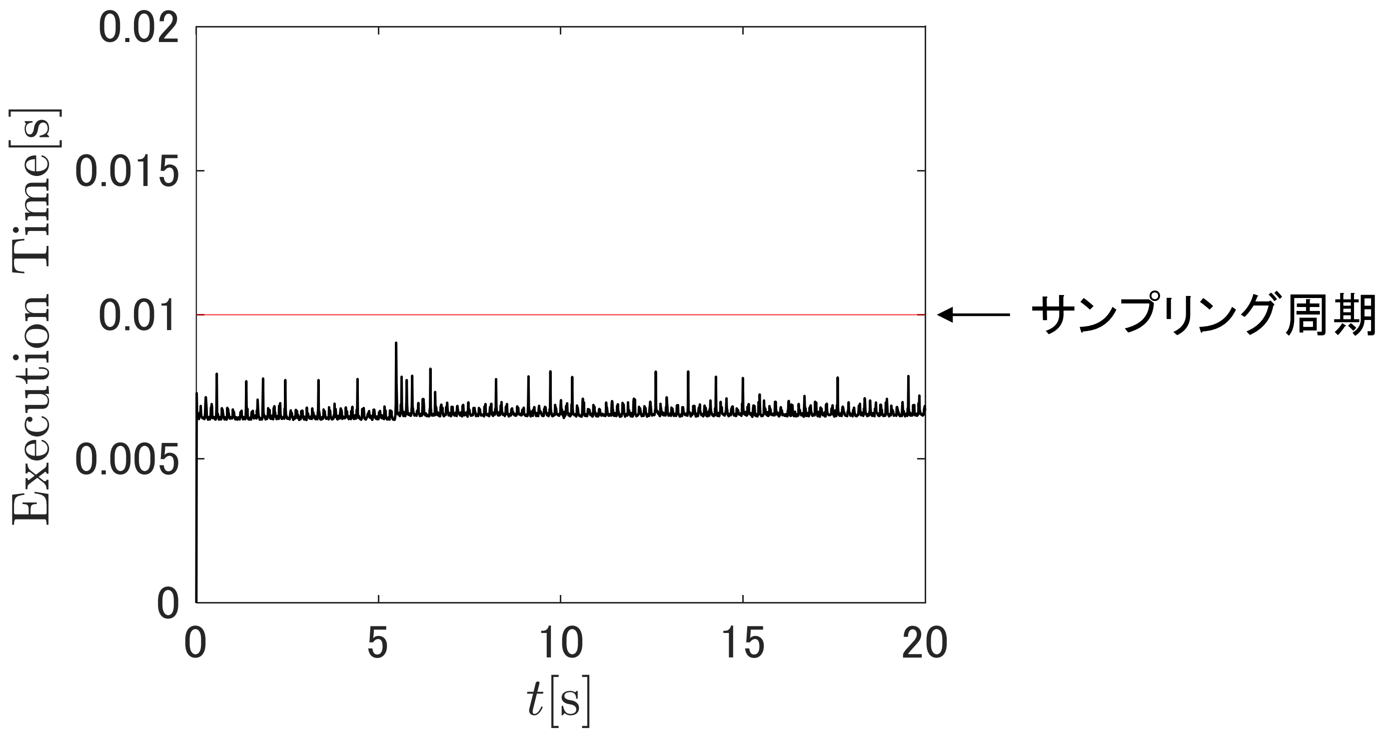
シミュレーション結果（故障検知パラメータ）



故障以降，検知パラメータの推定値が真値に追従
（センサ故障の検知への有用性を確認）

実時間処理の評価

- 粒子フィルタによる推定：複数の粒子（推定値の候補）を用意
- 課題：計算負荷の増加 ▶ GPU搭載のPCによる計算時間の評価



推定に要する計算時間がサンプリング周期未満（実時間処理を確保）

対象の数学モデルの構築と同定

・故障検知アルゴリズム：
　対象の数学モデルを用いて推定

$$x(k) = f(x(k-1), \theta\ , u(k), v(k))$$

▼

・課題：実システムでは同定（推定）が困難な
　　　非線形要素（例：摩擦，不感帯）を含む

▼

展望：
・NNなどの学習手法を適用
・学習の精度を保証するため，線形要素など
　モデル化可能な部分を除いた非線形要素のみを学習

おわりに

OKAYAMA UNIV.

まとめ

- 遠隔型の故障検知アルゴリズムの検討
- 粒子フィルタによる検知パラメータの推定
- 故障検知の実験環境の構築，シミュレーションによる検証

今後の展望

- 対象の非線形要素の同定（推定）
- 実機実験による検証