

# QoS Controlled NCS の安定化について

藤田研究室          米村 大輔

平成 18 年 1 月 30 日

## 1 はじめに

本レポートでは, QoS Controlled NCS について述べ, システムを安定化させる十分条件を提案する. さらに今後の課題について述べる.

## 2 QoS Controlled NCS

QoS(Quality of Service) Controlled NCS を Fig. 1 に示す. Decision Maker により Controller と QoS Network

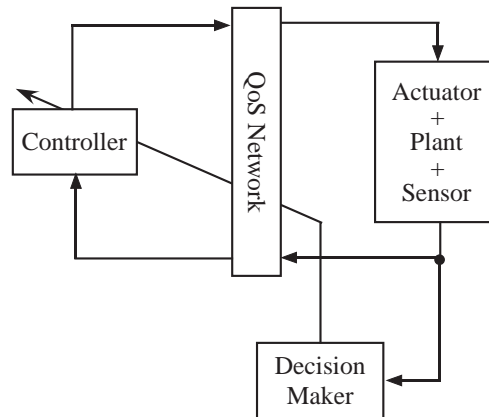


Fig.1: QoS Controlled NCS

をスイッチする. つまり, Fig. 1 に示されたシステムはハイブリッドシステム (Hybrid System) である.

使用する Network によってむだ時間が異なり, Packet dropout の頻度も変わる. 高価な Network と安価な Network を比較した場合, 高価な Network を使用すればむだ時間は短くなり, Packet dropout は減る. また, 安価な Network を使用すればむだ時間は大きくなり, Packet dropout は増えることが予想される. 以上のことから, Network も切り替えをすることを考える.

まず, むだ時間に注目する. Network に存在するむだ時間を固定とすると, 高価な Network を使用すればむだ時間は短くなり, 安価な Network を使用すればむだ時間は大きくなる. また, Network に存在するむだ時間を時変のむだ時間とした場合においても, 高級な Network を使用すればむだ時間の平均値は短くなり, 安価な Network を使用すればむだ時間の平均値は大きくなる. そこで, むだ時間とそれに対応するコストを例として Fig. 2 に示すように関係づける. そして, ある評価関数においてそのコストも最小にするように Decision Maker により切り替える.

⇒ 問題はどうか評価関数に組み込むか??

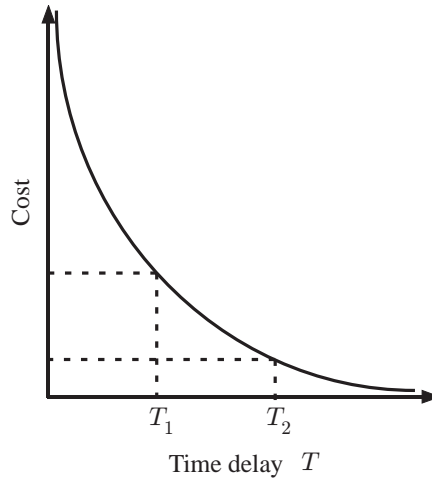


Fig.2: Delay-Cost

### 3 対象システムと問題の定式化

#### 3.1 対象システム

QoS Controlled NCS の簡単な例として Fig. 3 のようなシステムを考える. さらにむだ時間, コントローラを 2 つにしたもの考える (Fig. 4).

2 つのタイプのシステムを考える. まずタイプ 1, 2 のシステムはそれぞれむだ時間  $T_1, T_2$  が存在する Network を用いたシステムで, 次式で与えられるものとする.

$$\dot{x}(t) = Ax(t) + Bv(t) \quad (1)$$

$$(2)$$

ここで,  $x(t) \in \mathbb{R}^n$  は状態であり,  $v(t) \in \mathbb{R}^m$  は入力,  $A$  は  $n \times n$  の行列,  $B$  は  $n \times m$  の行列である.

#### 3.2 問題設定

type1, 2 の system をそれぞれ安定化する controller を設計し, さらにそれらをスイッチングした場合においてもシステム全体が安定となるように Decision Maker を構成することである. まず, タイプ 1, 2 それぞれ次式で与えられる state feedback controller を構成する.

$$u_1(t) = K_1x(t) \quad (3)$$

$$u_2(t) = K_2x(t) \quad (4)$$

Fig. 4 で考えているシステムでは input delay が存在するので, type1,2 それぞれにおいて controller で生成される制御入力と, system (1) の入力の間には次式の関係が成立する.

$$v(t) = u_1(t - T_1) \quad (5)$$

$$v(t) = u_2(t - T_2) \quad (6)$$

したがって, type1,2 それぞれにおいて, 得られる closed loop は

$$\dot{x}(t) = f_1(t) = Ax(t) + K_1x(t - T_1) \quad (7)$$

$$\dot{x}(t) = f_2(t) = Ax(t) + K_2x(t - T_2) \quad (8)$$

$$(9)$$

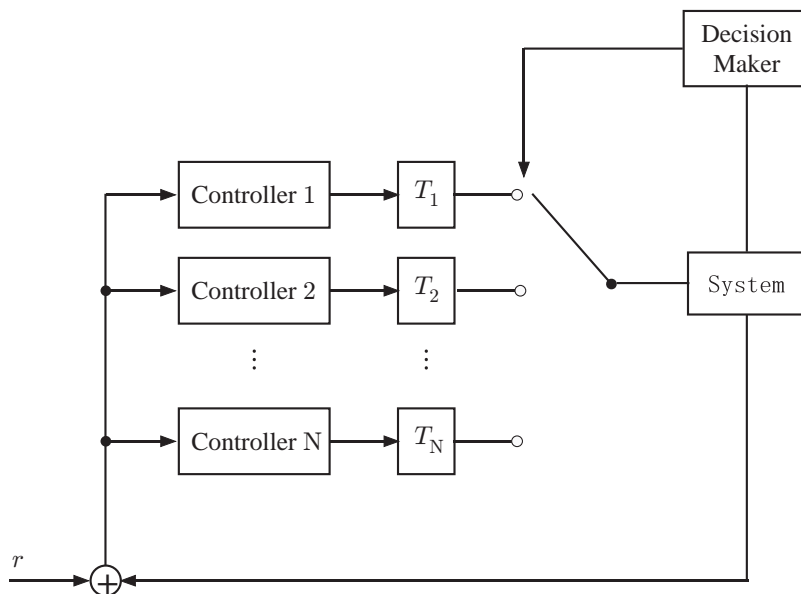


Fig.3: QoS Controlled NCS の例

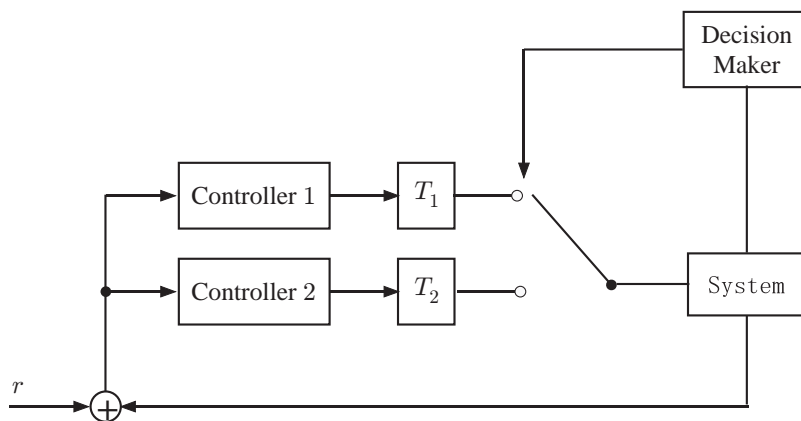


Fig.4: QoS Controlled NCS の例 (2Delay)

## 4 安定化コントローラの設計と Decision Maker の構成

### 4.1 むだ時間に依存しない安定化コントローラの設計

タイプ 1, 2 それぞれに対して次の Lyapunov Krasovskii Functional を用いてそれぞれの閉ループシステムを安定化させるコントローラを設計する.

$$V_1(x) = x(t)^T P_1^{-1} x(t) + \int_{-T_1}^0 x^T(t+\theta) P_1^{-1} Q_1 P_1^{-1} x(t+\theta) d\theta \quad (10)$$

$$V_2(x) = x(t)^T P_2^{-1} x(t) + \int_{-T_2}^0 x^T(t+\theta) P_2^{-1} Q_2 P_2^{-1} x(t+\theta) d\theta \quad (11)$$

コントローラの設計方法は文献 [1] を参照.

文献 [1] よりタイプ 1, 2 それぞれを安定化させるコントローラ (3), (4) が設計できたとする. しかし, これらのコントローラを切り替えたとしてもシステム全体が安定になるとは限らない (Fig. 5, 6).

**proposition1** Decision Maker を次のように構成することでシステム全体を安定化させることができる.

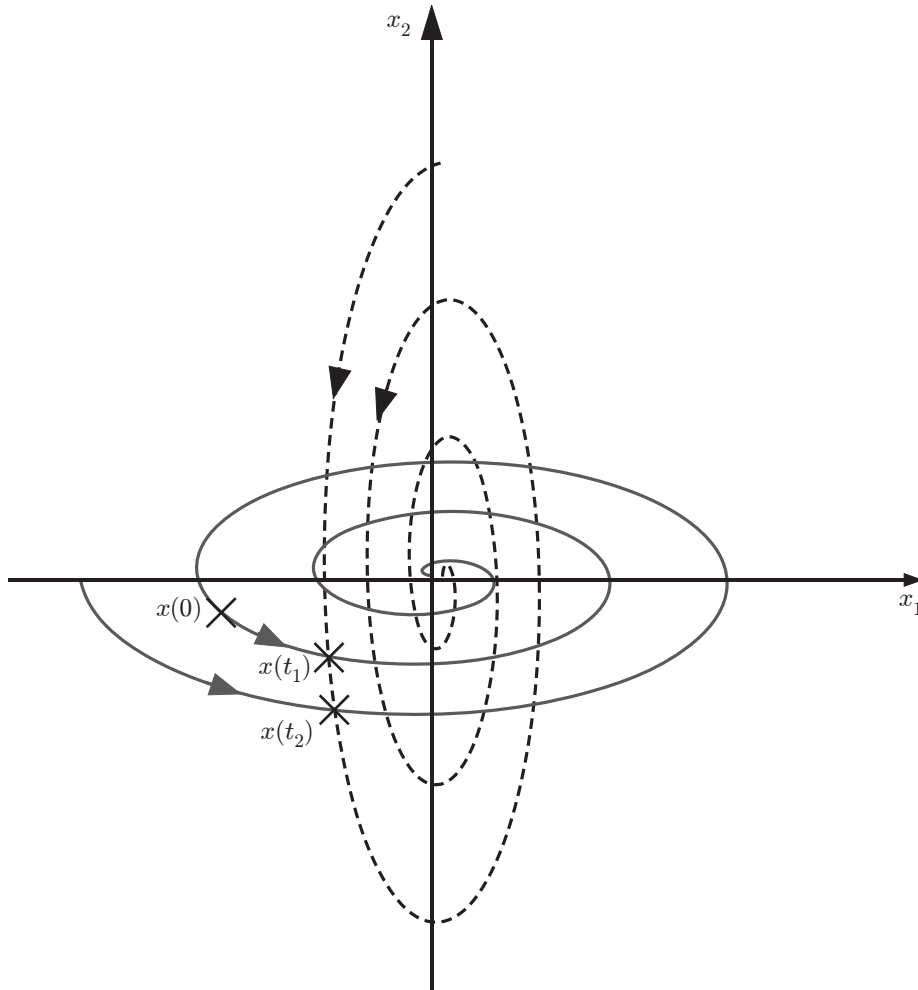


Fig.5: Hybrid System が不安定となる例

- $L_{f_1} V_2 = \frac{\partial V_2}{\partial x} f_1 < 0$  の時にタイプ 1 からタイプ 2 へ切り替える
- $L_{f_2} V_1 = \frac{\partial V_1}{\partial x} f_2 < 0$  の時にタイプ 2 からタイプ 1 へ切り替える

## 5 おわりに

前節で述べた条件は保守的な十分条件であるため今後改良が必要である。また、今後は Decision Maker の構成である評価関数を最小にする (最適とする) ようにコントローラを切り替えることを考えた。

また安定となる場合においても, Fig. 7, 82 の場合があると考えられる。さらに Fig. 9 のような場合においても安定となる。したがって, proposition1 以外にも安定とする方法がある。

## 参考文献

- [1] 米村, 東, “非線形無駄時間システムに対する安定化コントローラ的设计法,” 制御理論シンポジウム, 2005.

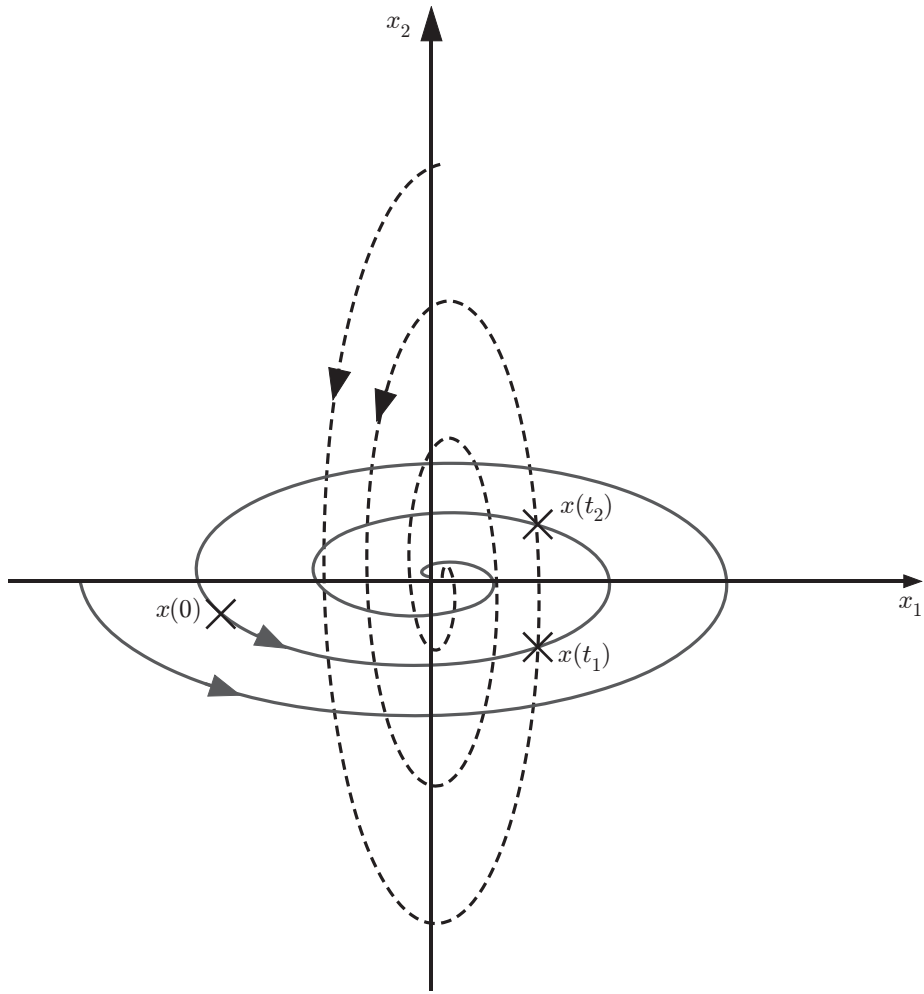


Fig.6: Hybrid System が安定となる例

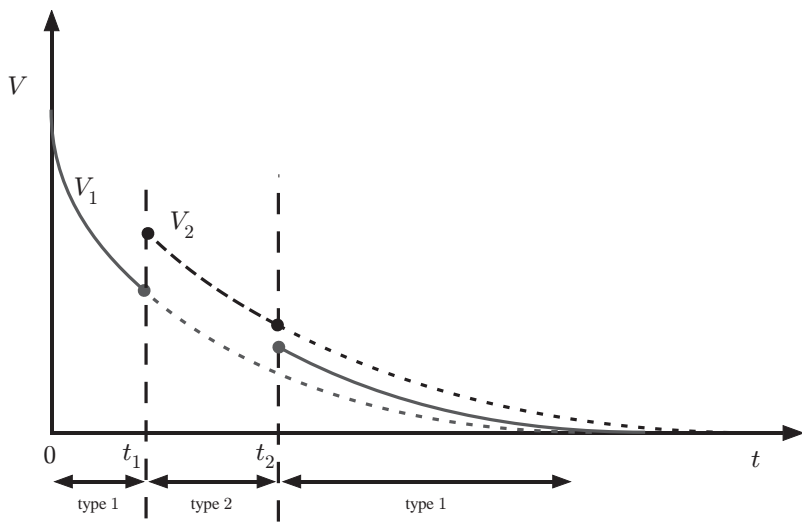


Fig.7: Hybrid System が安定となる例 1

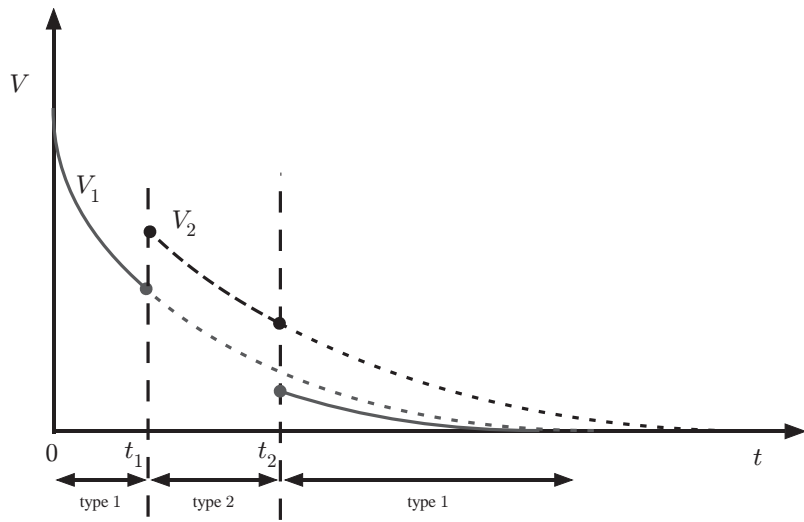


Fig.8: Hybrid System が安定となる例 2

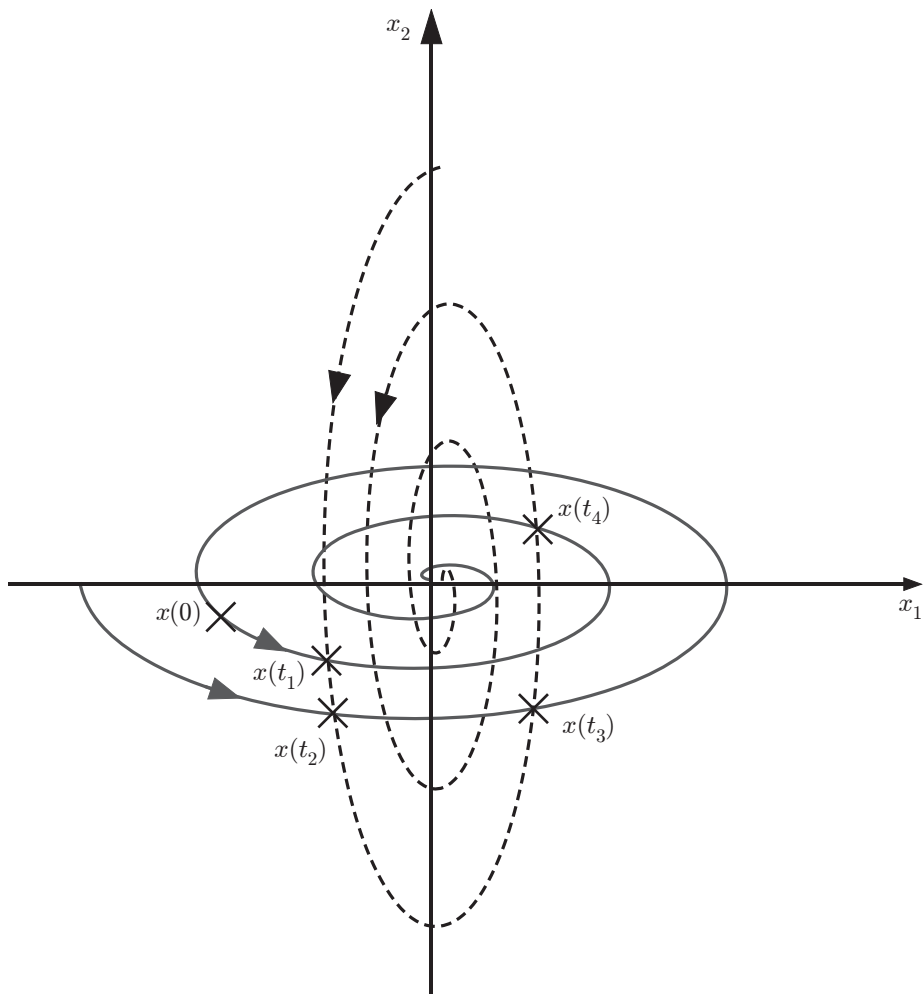


Fig.9: Hybrid System が安定となる例 1