

# リアルワールド応用知能集積システムのサンプリング周期の導出

○坂井勇太 亀山充隆 (東北大学)

## Derivation of a Sampling Period of an Intelligent Integrated System for Real-World Application

\*Yuta Sakai, Michitaka Kameyama (Tohoku University)

**Abstract**— To develop real-world applications such as highly-safe intelligent vehicles and home-service robots, we need to find the methodology to determine their performance specification. Necessary conditions for success of the goal behavior are derived from modeling real-world states as a set of difference equations. Then, a sampling period is obtained by solving the equation. As examples of the applications, we consider an autonomous intelligent vehicle and a catching-robot system.

**Key Words:** Real-world application, System LSI, Performance specification, Sampling period

### 1 まえがき

高安全知能自動車やホビーロボット, ホームサービスロボットなど様々なリアルワールド応用知能集積システムをプロセッサ化するに当たって, その処理時間仕様を与える方法論が望まれている. 本稿では目標の状態を実現するための処理時間仕様を, 漸化式に基づいて与える方法論について提案する. 目標の状態を実現するための必要条件を求め, さらにそれを実現するための任意の時刻における必要条件, すなわち差分方程式を導き出し, 処理時間仕様を与える. 具体例として高安全知能自動車の一例である自律走行車とボールの軌道予測を行いながら動作する捕球ロボットを取り上げ, 具体的に例を示して処理時間仕様を与える方法論の有用性を示す.

### 2 リアルワールド応用のサンプリング周期の導出

リアルワールド応用知能集積システムのシステムモデルとしてサンプリング周期  $\Delta T$  ごとに同一処理を繰り返す, プロセッサ化に適した  $\Delta T$  サイクルベースシステム (Fig.1) を前提とする [1]. このシステムではサンプリング周期  $\Delta T$  を決定することによりプロセッサの処理時間仕様を与えることができるため, サンプリング周期  $\Delta T$  を与える方法を考える.

Fig.2 はリアルワールド応用知能集積システムの状態の時系列を表したものである. 時刻  $T_{i+m}$  において目標の状態を実現するものとし,  $T_{i+1} = T_i + \Delta T$  である.  $E_i$  は時刻  $T_i$  における環境も含めた状態変数であり, 時間により変化する状態を数値や関数で表したものである.  $f_i$  は時刻  $T_i$  における目標の状態を実現するために必要な計画すべきパラメータである. 最終的に目標の状態を実現する時刻  $T_{i+m}$  でのパラメータを  $f_{i+m}$  とおくと  $\Delta T$  前のパラメータ  $f_{i+m-1}$  は  $f_{i+m}$  と  $E_{i+m-1}$  を用

いて表すことができる. 同様にして  $f_{i+m-2}$  は  $f_{i+m-1}$  と  $E_{i+m-2}$  を用いて表すことができる. このようにして, 任意の時刻間におけるパラメータの関係, すなわち差分方程式を導き出すことができ, これを解くことによってサンプリング周期を導出できる. ただし,  $E_i$  の自由度が大きくて予測ができず, 数値や関数等で表せないときは状況を限定する必要がある.

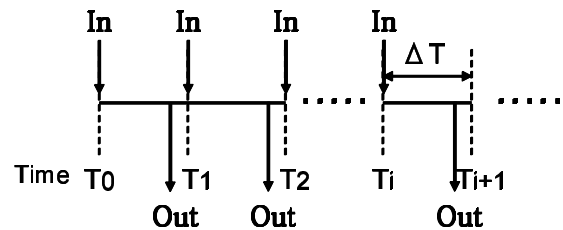


Fig. 1: Iteration with a sampling period  $\Delta T$

State Variable	.....	$E_i$	$E_{i+1}$	.....	$E_{i+m-1}$	$E_{i+m}$
Planning Parameter	.....	$f_i$	$f_{i+1}$	.....	$f_{i+m-1}$	$f_{i+m}$
Time	.....	$T_i$	$T_{i+1}$	.....	$T_{i+m-1}$	$T_{i+m}$

Fig. 2: State variable and a planning parameter in real-world application

### 3 自律走行車のサンプリング周期の導出

#### 3.1 自律走行システム

高安全知能自動車 [2] の一例である静的環境下における自律走行車を例にしてサンプリング周期を実際に求めてみる [3]. 自律走行システムとして、サンプリング周期  $\Delta T$  毎にカメラによる 3 次元環境情報の取得、軌道、速度の決定を行う離散時間処理を考える. Fig.3 は時刻  $T_i$ ,  $T_{i+1}$  の自動車の状態を表しており,  $T_{i+1} = T_i + \Delta T$  である. また,  $V_i$  は時刻  $T_i$  における速度であり, 各  $\Delta T$  間は等加速度運動であるとする.  $S_i$  は時刻  $T_i$  において軌道が存在する領域の距離であり,  $S_i$  が自由度の大きい状態変数となる. 軌道が存在しない領域は危険であると考えて (つまり,  $S_i$  を用いて) 速度計画を行い, 速度計画の結果からサンプリング周期を導出する.

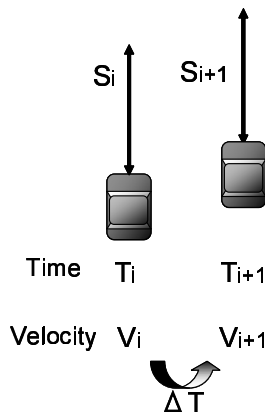


Fig. 3: Autonomous vehicle model

#### 3.2 速度計画

上記のように  $S_i$  を用いて時刻  $T_i$  で時刻  $T_{i+1}$  の速度  $V_{i+1}$  を決定する. しかし, 時刻  $T_{i+1}$  で安全であるように速度を決定してもそれ以降の時刻で安全が保証されなければ意味がない. そこで, Fig.4 のように時刻  $T_i$  以降軌道が全く増えないという最悪な場合に環境を限定し, その場合の安全な走行について考える. なぜならこの状況で安全であれば他のどの状況でも安全であるからである. 軌道が増えないため, 危険領域に侵入しないように時刻  $T_{i+1}$  以降は減速度  $\alpha$  以下で減速しなければならない. ここで,  $\alpha$  は危険でない許容最大減速度とする. 安全な領域がゼロとなる時刻を  $T_{i+m}$  とすると, 時刻  $T_{i+m}$  での安全条件は次のようになる.

$$C_{i+m} = S_i, \quad V_{i+m} = 0 \quad (1)$$

これは危険な領域の直前で停止することを意味し, 障害物に衝突して停止する場合は考えない. 次に,  $\Delta T$

前の時刻  $T_{i+m-1}$  について考えると, 各  $\Delta T$  間は等加速度運動であるので次式が成り立つ.

$$\begin{aligned} C_{i+m} - C_{i+m-1} &= \frac{V_{i+m} + V_{i+m-1}}{2} \Delta T \\ V_{i+m-1} - V_{i+m} &\leq \alpha \Delta T \end{aligned} \quad (2)$$

式 (1), (2) より時刻  $T_{i+m-1}$  での安全条件を求めることができる. 同様にして時間をさかのぼり時刻  $T_{i+1}$  の安全条件を求めることによって, 速度を次のように定式化できる. ただし, 連続と離散の関係により式 (1) を  $C_{i+m} = S_i - \frac{\alpha \Delta T^2}{2}$  とする.

$$\begin{aligned} 0 \leq V_{i+1} \leq & \frac{1}{2} \left\{ \sqrt{\alpha^2 \Delta T^2 + 8\alpha \left( S_i - \frac{\alpha \Delta T^2}{2} - \frac{V_i \Delta T}{2} \right)} \right. \\ & \left. - \alpha \Delta T \right\} \end{aligned} \quad (3)$$

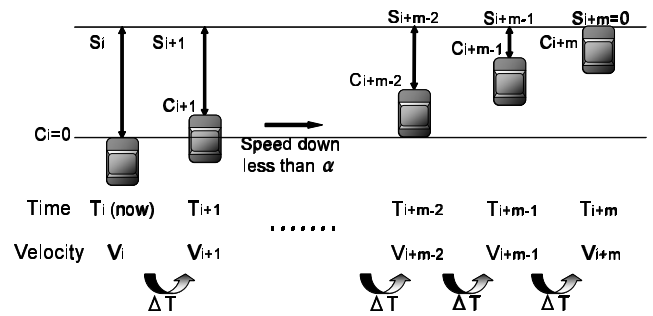


Fig. 4: Safe condition at the worst case

#### 3.3 サンプリング周期の導出

式 (3) を  $\Delta T$  について解くと次のようになる.

$$0 \leq \Delta T \leq \frac{1}{2\alpha} \left\{ \sqrt{(V_i + V_{i+1})^2 + 4(2\alpha S_i - V_{i+1}^2)} - (V_i + V_{i+1}) \right\} \quad (4)$$

ここで, サンプリング周期と速度の関係について述べる. サンプリング周期  $\Delta T$  が大きくなるほど  $\Delta T$  間に走行する距離が増加する, すなわち安全な領域が減少するため減速度を大きくしなければならない. しかし, 減速度にも上限があるため速度を小さくする必要が生じる. そこで実際の最高速度を設定し, 最高速度に達しても安全となるようにサンプリング周期  $\Delta T$  を決定する. これは式 (4) の右辺の最小値を求めることと等価である.  $V_i = V_{i+1} = V_{max}$  (最高速度) とおくと, 式 (4) よりサンプリング周期  $\Delta T$  は次式で与えられる.

$$0 \leq \Delta T \leq \frac{1}{\alpha}(\sqrt{2\alpha S_i} - V_{max}) \quad (5)$$

## 4 捕球ロボットのサンプリング周期の導出

### 4.1 誤差を考慮したシステムモデル

予測誤差を考慮したシステムとして捕球ロボットシステム (Fig.5) を取り上げる [4]. ここで,  $T_1 = T_0 + \Delta T$  である. このシステムではボールの次元座標をステレオビジョンにより計測し, 軌道予測を行う. 予測軌道に基づきロボットマニピュレータを制御する一連の処理をサンプリング周期  $\Delta T$  ごとに繰り返しながら, 捕球を行う.

誤差は通常データ数に依存すると考えられるので,  $\Delta T$  の関数としてモデル化できる. さらに, 捕球成功の条件に誤差モデルを適用することにより  $\Delta T$  を導出する. 議論を簡単にするため, マニピュレータは Y 軸上を移動する自由度とする. また, 風などはなくボールは放物線を描くものとする.

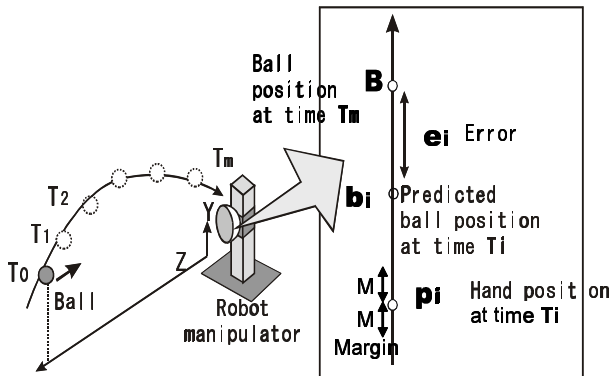


Fig. 5: Ball-catching robot system

### 4.2 サンプリング周期の導出

捕球成功のためのハンド位置の条件の定式化を行う. ボールは時刻  $T_0$  にマニピュレータに向かって投げられ, 時刻  $T_m$  において Y 軸上に到達する. 時刻  $T_i (0 \leq i < m)$  におけるハンドの位置を  $p_i$ , マニピュレータの移動速度を  $V$  とする. また, ボール到来位置の時刻  $T_i$  における予測値を  $b_i$ , 真のボール到来位置を  $B$  とする. ボールが Y 軸上に到達したときに, ボールがハンドの中心から距離  $M (M: \text{捕球マージン})$  以内であれば捕球は成功する. したがって, 時刻  $T_m$  におけるハンド位置の条件は次式で与えられる.

$$|p_m - B| < M \quad (6)$$

この条件が満たされるように時刻をさかのぼっていくと, 時刻  $T_i$  における捕球成功のためのハンド位置に関する必要条件は次式で与えられる.

$$|p_i - B| < M + V(T_m - T_i) \quad (7)$$

この条件は, ハンドが真のボール到来位置から残り時間で移動可能な距離以内に存在しなければ, 捕球を成功できないことを意味している. つぎに, ボールの到来位置の時刻  $T_i$  における予測誤差  $e_i$  が満たすべき条件を求める. 制御方式としては, ボールの予測到来位置にハンドを動かす. すなわち, 次式のように, 時刻  $T_i$  でのボール予測位置が時刻  $T_{i+1}$  でのハンドの位置になるように制御する.

$$p_{i+1} = b_i \quad (8)$$

このように, 制御値とハンドの位置の関係を定式化することにより, ハンド位置と誤差との関係を次式のように求めることができる.

$$e_i = |b_i - B| = |p_{i+1} - B| \quad (9)$$

ここで, 式 (7) より  $|p_{i+1} - B| < M + V(T_m - T_i - \Delta T)$  であるので, 誤差の条件は次式のように整理される.

$$e_i < M + V(T_m - T_i - \Delta T) \quad (10)$$

以上, 捕球成功のための任意の時刻  $T_i$  における誤差の条件を求めた. これを  $\Delta T$  について解くと,

$$\Delta T < \frac{M - e_i + V(T_m - T_i)}{V} \quad (11)$$

ここで, マニピュレータの動作速度が遅く捕球成功のための誤差の最大値が存在することから, 誤差の最大値  $e_{max}$  を実験的に決定することでサンプリング周期  $\Delta T$  を導出する.  $e_{max}$  の一例として, 次のような関数で与える.

$$e_{max} = \frac{C_1 \Delta T}{(T_i + C_2) + C_3} \quad (12)$$

式 (11), (12) より, 最小の  $\Delta T$  を求めることで  $\Delta T$  が次式のように与えられる.

$$\Delta T < \min \left[ \frac{M + V(T_m - T_i) - C_3}{\frac{C_1}{T_i + C_2} + V} \right] \quad (13)$$

## 5 サンプリング周期の評価

目標の状態を実現するために必要なパラメータの条件から差分方程式を求め、自律走行車と捕球ロボットの2例についてサンプリング周期を導出した。この2例からリアルワールド応用知能集積システムのサンプリング周期を一般的に導出する方法について考察する。Fig.6は自律走行車、捕球ロボットシステムにおいて $V_{i+1}$ ,  $e_i$ ,  $\Delta T$ 以外の変数がある一定の値にした場合の $V_{i+1}$ と $\Delta T$ ,  $e_i$ と $\Delta T$ との関係をグラフで表したものである。自律走行システムにおいては $V_i = 60(\text{km/h})$ ,  $\alpha = 10(\text{km/h} \cdot \text{s})$ ,  $S_i = 50(\text{m})$ に、捕球ロボットシステムにおいては $M = 10(\text{cm})$ ,  $V = 20(\text{cm/s})$ ,  $T_m = 5(\text{s})$ ,  $T_i = 2(\text{s})$ にそれぞれ値を設定している。点線が $V_{i+1}$ を、実線が $e_i$ を表している。

Fig.6より、速度や誤差がともにサンプリング周期 $\Delta T$ に依存していることがわかる。自律走行車と捕球ロボットの2例について、差分方程式を求め、サンプリング周期 $\Delta T$ と依存関係にあるパラメータ(速度や誤差)の最大値を設定することによってサンプリング周期を導出できる。すなわち、式(4), (11)の右辺の最小値が最も厳しい条件となることに着目している。以上のように、サンプリング周期を支配するパラメータの存在に着目することが重要である。

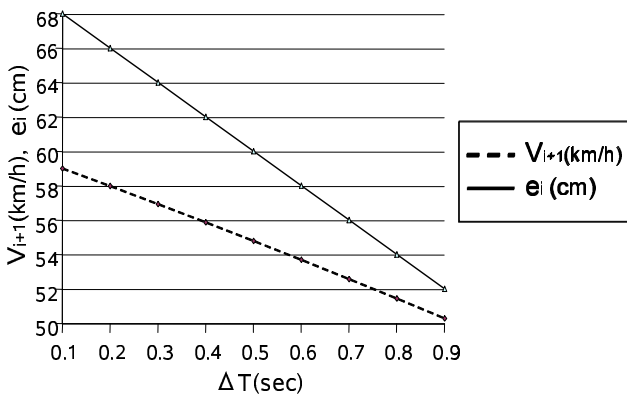


Fig. 6: Effect for a sampling period  $\Delta T$

## 6 むすび

リアルワールド応用知能集積システムの処理時間を与えるサンプリング周期を導出するため、目標の状態を実現する時刻での必要条件を求め、さらに任意の時刻での必要条件から差分方程式を求める方法を提案した。また、差分方程式からサンプリング周期 $\Delta T$ を求める方法についても、サンプリング周期 $\Delta T$ を支配するパラメータに着目することが重要であることを述べ

た。今回の具体例は自律走行車と捕球ロボットの2つであるが、今後は他のリアルワールド応用知能集積システムについても評価する必要がある。

## 参考文献

- [1] 亀山 充隆, 張山 昌論: リアルワールド応用知能集積システムの展望, 計測と制御, Vol.40, No.12, pp. 842-843, 2001.
- [2] 張山 昌論, 亀山 充隆: 高性能軌道計画VLSIプロセッサと高安全知能自動車への応用, 電子情報通信学会, 信学技報, pp. 26-31, 2000.
- [3] 坂井 勇太, 亀山 充隆: 高安全知能自動車のためのシステムインテグレーション, 平成16年度電気関係学会東北支部連合大会, 116, pp. 309, 2004.
- [4] M. Kameyama, M. Hariyama: Design Methodology for Human-Oriented Intelligent Integrated Systems, Interdisciplinary Information Sciences, Vol.7, No.2, pp. 280-283, 2001.