

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2013-225284

(P2013-225284A)

(43) 公開日 平成25年10月31日(2013.10.31)

(51) Int.Cl.	F I	テーマコード (参考)
<b>G05D 3/12 (2006.01)</b>	G05D 3/12 305V	5H004
<b>G05B 11/36 (2006.01)</b>	G05D 3/12 306Z	5H303
<b>G05B 13/02 (2006.01)</b>	G05B 11/36 501A	
	G05B 13/02 B	

審査請求 未請求 請求項の数 20 O L 外国語出願 (全 161 頁)

(21) 出願番号	特願2012-226827 (P2012-226827)	(71) 出願人	512264806
(22) 出願日	平成24年10月12日 (2012.10.12)		ラインストリーム テクノロジーズ
(31) 優先権主張番号	13/451,924		LINESTREAM TECHNOLOGIES
(32) 優先日	平成24年4月20日 (2012.4.20)		アメリカ合衆国 44113 オハイオ州
(33) 優先権主張国	米国 (US)		クリーブランド スート100 ナイン
(31) 優先権主張番号	13/474,919		ス ストリート ダブリュー. 1468
(32) 優先日	平成24年5月18日 (2012.5.18)	(74) 代理人	100108062
(33) 優先権主張国	米国 (US)		弁理士 日向寺 雅彦
		(74) 代理人	100168332
			弁理士 小崎 純一
		(74) 代理人	100146592
			弁理士 市川 浩
		(74) 代理人	100159709
			弁理士 本間 惣一

最終頁に続く

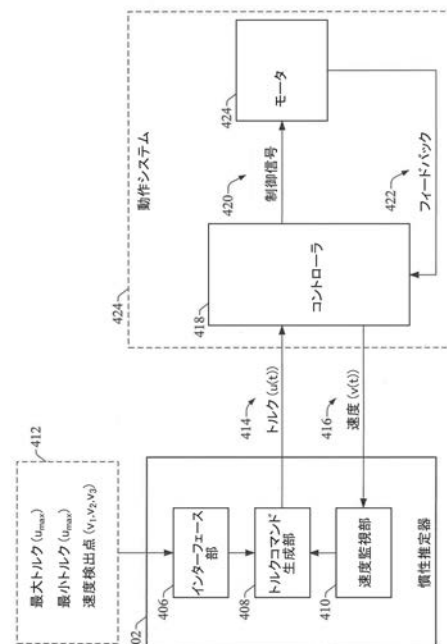
(54) 【発明の名称】 動作システムのパラメータ推定法、そのパラメータ推定システム、動作プロファイルの生成方法およびその生成システム

(57) 【要約】 (修正有)

【課題】 制御される機械システムの慣性および摩擦係数を推定するシステムおよび方法を提供する。

【解決手段】 慣性推定器 402 は、テストシーケンスの間、時間に対して連続的に変化するトルクコマンド信号を生成する。時間的に変化するトルクコマンド信号に回答する動作システムの速度は、テストシーケンスの間に測定され記録する。そして、慣性推定器は、動作システムに送られたトルクコマンドデータ、および、測定された速度データに基づいて、動作システムの慣性および摩擦係数のいずれかを推定する。動作制御システムにおける 2 点間の移動の軌跡を制御するための制約に基づいて時間的に最適化された動作プロファイルも生成する。また、プロファイル生成部は、動作プロファイルの少なくとも 1 つのセグメントにおいて、時間に対して連続的に変化するジャーク基準を含む S T 曲線の動作プロファイルを計算し、スムーズな時間的に最適化された軌跡を創り出す。

【選択図】 図 4



## 【特許請求の範囲】

## 【請求項 1】

時間に対して連続的に変化するトルクコマンド信号を生成し、  
前記トルクコマンド信号に応答する動作システムの速度を表す、動作デバイスの速度データを測定し、

前記速度データおよび前記トルクコマンド信号の少なくとも一部に基づいて、前記動作システムの慣性および摩擦係数の少なくとも 1 つを決定する動作システムのパラメータ推定法。

## 【請求項 2】

前記決定は、前記速度データの時間領域に対する 1 以上の積分、および、前記トルクコマンド信号の前記時間領域に対する 1 以上の積分の少なくとも一部に基づいて行われる請求項 1 記載の動作システムのパラメータ推定法。

10

## 【請求項 3】

前記トルクコマンド信号の生成は、予め定められたテストシーケンスにしたがったトルクコマンド信号の調整を含み、

前記調整は、予め定められた速度検出点に到達した前記動作システムの前記速度に応答した前記トルクコマンド信号の方向もしくは変化のレートの少なくとも 1 つを変化させる請求項 1 または 2 に記載の動作システムのパラメータ推定法。

## 【請求項 4】

前記決定は、前記時間領域の第 1 セグメントを加速フェーズ、前記時間領域の第 2 セグメントを減速領域とし、

20

$U_{acc}$  および  $V_{acc}$  を得るために前記トルクコマンド信号および前記速度データを前記加速フェーズにおいてそれぞれ積分し、

$U_{dec}$  および  $V_{dec}$  を得るために前記トルクコマンド信号および前記速度データを前記減速フェーズにおいてそれぞれ積分し、

前記慣性および前記摩擦係数の少なくともいずれか 1 つを  $U_{acc}$ 、 $V_{acc}$ 、 $U_{dec}$  および  $V_{dec}$  の関数として決定する請求項 2 または 3 に記載の動作システムのパラメータ推定法。

ここで、

## 【数 57】

30

$$U_{acc} = \int u_{acc}(t)$$

## 【数 58】

$$V_{acc} = \int v_{acc}(t)$$

## 【数 59】

$$U_{dec} = \int u_{dec}(t)$$

40

## 【数 60】

$$V_{dec} = \int v_{dec}(t)$$

$u_{acc}(t)$  は、前記加速フェーズに対応する前記トルクコマンド信号の一部であり、

$v_{acc}(t)$  は、前記加速フェーズに対応する前記速度データの一部であり、

$u_{dec}(t)$  は、前記減速フェーズに対応する前記トルクコマンド信号の一部であり、

50

$v_{dec}(t)$  は、前記減速フェーズに対応する前記速度データの一部である。

【請求項 5】

前記慣性および前記摩擦係数の少なくともいずれか 1 つの前記決定は、前記関数にしたがって前記慣性を決定すること、および、前記関数にしたがって前記摩擦係数を決定すること、の少なくともいずれか 1 つを含む請求項 4 記載の動作システムのパラメータ推定法。

ここで、前記慣性を表す関数  $J$  は、

【数 6 1】

$$J = \frac{U_{dec} V_{acc} - U_{acc} V_{dec}}{\Delta v_{dec}(t) V_{acc} - \Delta v_{acc}(t) V_{dec}} \quad 10$$

であり、

前記摩擦係数を表す関数  $B$  は、

【数 6 2】

$$B = \frac{\Delta v_{dec}(t) U_{acc} - \Delta v_{acc}(t) U_{dec}}{\Delta v_{dec}(t) V_{acc} - \Delta v_{acc}(t) V_{dec}} \quad 20$$

であり、

$v_{acc}(t)$  は、前記加速フェーズの終了時における前記動作システムの速度と、前記加速フェーズの開始時における前記動作システムの速度と、の差であり、

$v_{dec}(t)$  は、前記減速フェーズの終了時における前記動作システムの速度と、前記減速フェーズの開始時における前記動作システムの速度と、の差である。

【請求項 6】

前記慣性および前記摩擦係数の少なくともいずれか 1 つに基づいて、前記動作システムの制御利得係数を決定する請求項 1 ~ 5 のいずれか 1 つに記載の動作システムのパラメータ推定法。

【請求項 7】

メモリと、

前記メモリに保存されたコンピュータが実行可能な要素を実行するプロセッサと、を備え、

前記コンピュータが実行可能な要素は、

テストシーケンスの間、時間に対して連続的に変化するトルクコマンド信号を生成するトルクコマンド生成部と、

前記トルクコマンド信号に応答する動作システムの速度を時間に対して表した速度データを取得する速度監視部と、

前記トルクコマンド信号および前記速度データに基づいて、前記動作システムの慣性を推定する慣性演算部、および、前記トルクコマンド信号および前記速度データに基づいて、前記動作システムの摩擦係数を推定する摩擦係数演算部、の少なくともいずれか 1 つと、

を含む動作システムのパラメータ推定システム。

【請求項 8】

前記テストシーケンスの継続時間の少なくとも一部である時間領域において、前記慣性演算部および前記摩擦係数演算部の少なくともいずれか 1 つは、前記時間領域における速度データの 1 以上の積分、および、前記時間領域におけるトルクコマンド信号の 1 以上の積分の少なくとも一部に基づいて、前記慣性または前記摩擦係数をそれぞれ推定する請求項 7 記載の動作システムのパラメータ推定システム。

【請求項 9】

前記トルクコマンド生成部は、前記テストシーケンスのそれぞれのフェーズに関連して

10

20

30

40

50

予め定められた命令に基づいたトルク関数  $u(t)$  にしたがってトルクコマンド信号を制御し、

前記それぞれのフェーズは、それぞれに定められた速度検出点の値に達した前記動作システムの前記速度に応答して開始される請求項 7 または 8 に記載の動作システムのパラメータ推定システム。

【請求項 10】

前記慣性演算部および前記摩擦係数演算部の少なくともいずれか 1 つは、 $U_{acc}$ 、 $V_{acc}$ 、 $U_{dec}$  および  $V_{dec}$  の関数として前記慣性を推定する請求項 8 または 9 に記載の動作システムのパラメータ推定システム。

10

ここで、

【数 57】

$$U_{acc} = \int u_{acc}(t)$$

【数 58】

$$V_{acc} = \int v_{acc}(t)$$

【数 59】

$$U_{dec} = \int u_{dec}(t)$$

20

【数 60】

$$V_{dec} = \int v_{dec}(t)$$

であり、

$u_{acc}(t)$  は、前記テストシーケンスの前記加速フェーズに対応する前記トルクコマンド信号の一部であり、

30

$v_{acc}(t)$  は、前記加速フェーズに対応する前記速度データの一部であり、

$u_{dec}(t)$  は、前記テストシーケンスの前記減速フェーズに対応する前記トルクコマンド信号の一部であり、

$v_{dec}(t)$  は、前記減速フェーズに対応する前記速度データの一部である。

【請求項 11】

前記慣性演算部は、前記関数に基づいて前記慣性を推定する請求項 10 に記載の動作システムのパラメータ推定システム。

ここで、前記慣性を表す関数  $J$  は、

【数 61】

$$J = \frac{U_{dec} V_{acc} - U_{acc} V_{dec}}{\Delta v_{dec}(t) V_{acc} - \Delta v_{acc}(t) V_{dec}}$$

40

であり、

$v_{acc}(t)$  は、前記加速フェーズの終了時における前記動作システムの速度と、前記加速フェーズの開始時における前記動作システムの速度と、の差であり、

$v_{dec}(t)$  は、前記減速フェーズの終了時における前記動作システムの速度と、前記減速フェーズの開始時における前記動作システムの速度と、の差である。

【請求項 12】

前記摩擦係数演算部は、前記関数に基づいて前記摩擦係数を推定する請求項 10 に記載の

50



動作システムのパラメータ推定システム。

ここで、前記慣性を表す関数  $B$  は、

【数 6 2】

$$B = \frac{\Delta v_{dec}(t)U_{acc} - \Delta v_{acc}U_{dec}}{\Delta v_{dec}(t)V_{acc} - \Delta v_{acc}(t)V_{dec}}$$

であり、

$v_{acc}(t)$  は、前記加速フェーズの終了時における前記動作システムの速度と、前記加速フェーズの開始時における前記動作システムの速度と、の差であり、

10

$v_{dec}(t)$  は、前記減速フェーズの終了時における前記動作システムの速度と、前記減速フェーズの開始時における前記動作システムの速度と、の差である。

【請求項 1 3】

前記慣性および前記摩擦係数の少なくともいずれか 1 つの関数として、少なくとも 1 つの制御利得係数を生成するチューニング部をさらに備えた請求項 7 ~ 12 のいずれか 1 つに記載の動作システムのパラメータ推定システム。

【請求項 1 4】

動作デバイスの目標ポジションおよび目標速度の少なくともいずれか 1 つを指示する設定点を受信し、

前記目標ポジションおよび前記目標速度の少なくともいずれか 1 つに、前記動作デバイスを移行させるための動作プロファイルを生成し、

20

前記動作プロファイルは、その少なくとも 1 つのセグメントに対し、時間の関数として連続的に変化するジャーク基準を定める動作プロファイルの生成方法。

【請求項 1 5】

前記動作プロファイルの前記生成は、その少なくとも 1 つのセグメントに対して、時間の関数として、加速度基準、速度基準およびポジション基準のうちの少なくとも 1 つを定めることを含む請求項 1 4 記載の動作プロファイルの生成方法。

【請求項 1 6】

前記生成は、前記動作プロファイルのそれぞれのセグメントに対して、前記ジャーク基準、速度基準およびポジション基準のうちの少なくとも 1 つを計算し、

30

前記動作プロファイルの前記それぞれのセグメントの継続時間を計算し、

前記継続時間を、動作コントローラのサンプリング時間のそれぞれに最も近い倍数に修正し、

前記修正された継続時間を用いて、前記ジャーク基準、前記加速度基準、前記減速度基準および前記速度基準のうちの少なくとも 1 つを再計算する請求項 1 4 または 1 5 に記載の動作プロファイルの生成方法。

【請求項 1 7】

前記生成は、

【数 6 3】

$$t_1^3(t_1+t_2) = t_4^3(t_4+t_5)$$

40

【数 6 4】

$$t_1 = \frac{3A}{2Je}$$

【数 6 5】

$$t_4 = \frac{3D}{2I}$$

【数 6 6】

$$t_2 = \frac{V}{A} - \frac{3A}{2Je}$$

【数 6 7】

$$t_5 = \frac{V}{D} - \frac{3D}{2I}$$

【数 6 8】

$$t_3 = \frac{P}{V} - \frac{3A}{4Je} - \frac{V}{2A} - \frac{3D}{4I} - \frac{V}{2D}$$

10

【数 6 9】

$$\frac{A^2}{D^2} = \frac{Je^3}{I^3}$$

【数 7 0】

$$V \geq \frac{3A^2}{2Je} = \frac{3}{2I\rho^{\frac{4}{3}}} D^2$$

20

【数 7 1】

$$V \geq \frac{3}{2I} D^2$$

【数 7 2】

$$\frac{P}{V} \geq \frac{3}{4I} \rho^{\frac{1}{3}} D + \frac{\rho}{2} \frac{V}{D} + \frac{3}{4I} D + \frac{1}{2} \frac{V}{D}$$

30

【数 7 3】

$$\frac{P}{V} \geq \left( \rho^{\frac{1}{3}} + 1 \right) \frac{3}{4I} D + \frac{1}{2} \left( \rho + 1 \right) \frac{V}{D}$$

【数 7 4】

$$\frac{I}{Je} = \rho^{\frac{2}{3}}$$

の関係にしたがって、前記動作プロファイルを生成する請求項 14 ~ 16 のいずれか 1 つに記載の動作プロファイルの生成方法。

40

ここで、Pは前記動作デバイスのポジション、Jeは最大加速ジャーク、Iは最大減速ジャーク、Aは最大加速度、Dは最大減速度、Vは最大速度であり、

t<sub>1</sub>は、前記動作プロファイルにおける加速度が増加する段階および減少する段階の継続時間、t<sub>2</sub>は、前記動作プロファイルの加速度が一定の段階の継続時間、t<sub>3</sub>は、前記動作プロファイルの一定速度の段階の継続時間、t<sub>4</sub>は、前記動作プロファイルにおける減速度が増加する段階および減少する段階の継続時間、t<sub>5</sub>は、動作プロファイルの減速度が一定の段階の継続時間であり、

は、

【数 7 5】

$$\rho = \frac{D}{A}$$

で表される。

【請求項 1 8】

メモリと、

前記メモリに保存されたコンピュータが実行可能な要素を実行するプロセッサと、を備え、

前記コンピュータが実行可能な要素は、動作デバイスの目標ポジションまたは目標速度の受信に応答して、動作プロファイルを生成する動作プロファイル生成部を含み、 10

前記動作プロファイルは、その少なくとも 1 つの段階において、時間に対して連続的に変化するジャーク基準を有し、前記動作デバイスを前記目標ポジションまたは前記目標速度に移行させるための軌跡を定める動作プロファイルの生成システム。

【請求項 1 9】

前記動作プロファイル生成部は、少なくとも 1 つの定められた制約条件に依存する動作プロファイルを生成し、

前記少なくとも 1 つの定められた制約条件は、サンプリング時間、速度限界、加速度限界、ジャーク限界および減速度限界のうちの少なくとも 1 つを含む請求項 1 8 記載の動作プロファイルの生成システム。 20

【請求項 2 0】

前記動作プロファイル生成部は、

【数 6 3】

$$t_1^3(t_1+t_2) = t_4^3(t_4+t_5)$$

【数 6 4】

$$t_1 = \frac{3A}{2Je}$$

【数 6 5】

$$t_4 = \frac{3D}{2I}$$

【数 6 6】

$$t_2 = \frac{V}{A} - \frac{3A}{2Je}$$

【数 6 7】

$$t_5 = \frac{V}{D} - \frac{3D}{2I}$$

【数 6 8】

$$t_3 = \frac{P}{V} - \frac{3A}{4Je} - \frac{V}{2A} - \frac{3D}{4I} - \frac{V}{2D}$$

【数 6 9】

$$\frac{A^2}{D^2} = \frac{Je^3}{I^3}$$

10

20

30

40

【数 7 0】

$$V \geq \frac{3A^2}{2J_e} = \frac{3}{2I\rho^{\frac{4}{3}}} D^2$$

【数 7 1】

$$V \geq \frac{3}{2I} D^2$$

【数 7 2】

10

$$\frac{P}{V} \geq \frac{3}{4I} \rho^{-\frac{1}{3}} D + \frac{\rho}{2} \frac{V}{D} + \frac{3}{4I} D + \frac{1}{2} \frac{V}{D}$$

【数 7 3】

$$\frac{P}{V} \geq \left( \rho^{-\frac{1}{3}} + 1 \right) \frac{3}{4I} D + \frac{1}{2} \left( \rho + 1 \right) \frac{V}{D}$$

【数 7 4】

20

$$\frac{I}{J_e} = \rho^{\frac{2}{3}}$$

の関係にしたがって、前記動作プロファイルを生成する請求項 18 または 19 に記載の動作プロファイルの生成システム。

ここで、P は前記動作デバイスのポジション、J<sub>e</sub> は最大加速ジャーク、I は最大減速ジャーク、A は最大加速度、D は最大減速度、V は最大速度であり、

t<sub>1</sub> は、前記動作プロファイルにおける加速度が増加する段階および減少する段階の継続時間、t<sub>2</sub> は、前記動作プロファイルの加速度が一定の段階の継続時間、t<sub>3</sub> は、前記動作プロファイルの一定速度の段階の継続時間、t<sub>4</sub> は、前記動作プロファイルにおける減速度が増加する段階および減少する段階の継続時間、t<sub>5</sub> は、動作プロファイルの減速度が一定の段階の継続時間であり、

30

は、

【数 7 5】

$$\rho = \frac{D}{A}$$

で表される。

【発明の詳細な説明】

【技術分野】

【0001】

40

本発明の実施形態は、動作システムのパラメータ推定法、そのパラメータ推定システム、動作プロファイルの生成方法およびその生成システムに関し、機械システムにおける慣性の自動推定方法および動作プロファイルの生成方法、一般的な動作制御、特に、動作制御システムにおいてパラメータとして使用する慣性および摩擦係数の推定、および、制約条件に基づく時間的最適動作のプロファイル生成に関する。

【背景技術】

【0002】

自動化を適用する多くのケースでは、機械のポジションと速度を制御するために動作制御システムを採用する。そのような動作制御システムは、通常、コントローラの制御により動作する 1 つ以上のモータまたは類似した動作装置を含む。コントローラは、ユーザに

50

定義された制御アルゴリズムまたはプログラムにしたがって、ポジションおよび速度の制御命令をモータに送る。いくつかの動作制御システムは、閉ループ構成で動作し、コントローラは、目標ポジションに移動するか、または、目標速度（望ましい状態）に移行するようにモータに命令し、モータの動作状態を示すフィードバック情報を受信する。コントローラは、モータが目標ポジションもしくは目標速度に達したかどうかを決定するためにフィードバック情報を監視し、実際の状態と望ましい状態との間の誤差を補正するために制御信号を調整する。

#### 【0003】

動作制御システムの設計者は、動作速度とシステムの安定性との間の最適なトレードオフを追求する。例えば、機械部品を目標ポジションに高トルクで遷移させることをコントローラがモータに命令すれば、機械は、最初に、そのポジションと望ましいポジションとのズレを高速（すなわち、時間的に効率的な方法）で縮めるが、高トルクのために望ましいポジションを通過してしまうだろう。結果的に、コントローラは、機械を望ましいポジションに戻すために補正信号を出力しなければならない。動作システムを望ましいポジションに落ち着かせる前に、そのような相互作用が繰り返し実行され、望ましくない機械振動が生じるであろう。反対に、モータに低トルクで移動するように命令すると、初期状態の遷移の精度を高くし、機械振動を低減もしくは無くすることができであろう。しかし、望ましいポジションに機械を移動させるための時間が長くなる。理想的には、状態の遷移速度と、システムの安定性と、の間のトレードオフを最適化するために、コントローラの利得係数が選択されるべきである。コントローラの好適な利得係数の選択は、チューニングとして知られている。

#### 【0004】

所定の利得係数を有するコントローラの信号に対する制御される機械システムの応答は、慣性および摩擦係数を含めた機械システムの物理的な特性に依存する。慣性は、動作システムの加速および減速に対する抵抗を表す。摩擦係数は、ロータとシャフトとの間の摩擦のようなモータに見られる摩擦を表す。制御された機械システムの慣性および摩擦係数に対する正確な推定は、チューニング過程を簡略化し、システムの性能を向上させる。しかしながら、任意の機械システムに対してこれらのパラメータの正確な値を同定することは難しい。いくつかのケースでは、慣性は、モータの定格と、負荷を含めた部品の物理的データ（重さ、寸法など）と、に基づいた手計算を用いて推定される。そのような計算は、厄介であり時間がかかる。そして、これらの重要なパラメータの正確な値を得ることは難しい。

#### 【0005】

動作制御の他の態様では、ポジションもしくは速度の状態遷移を促すために動作プロファイルがしばしば用いられる。例えば、（制御アルゴリズムまたはユーザの要求にしたがって）コントローラが、動作システムが新しいポジションに移動またはその速度を変化させなければならないことを決定する時、動作システムをそのポジション／速度から目標ポジション／速度へ遷移させるためのポジションまたは速度の軌跡を計算しなければならない。その軌跡は、動作プロファイルとして言及され、動作プロファイルは、そのシステムが現在の状態から目標の状態に移動する時間の経過とともに動作システムの速度、加速度およびポジション、もしくは、それぞれを明らかにする。一旦、この動作プロファイルが計算されれば、コントローラは、プロファイルによって明らかにされた軌跡を通して動作システムを動かすために、動作プロファイルを適切な制御信号に変換する。

#### 【0006】

いくつかの用途では、動作プロファイルのさまざまなセグメント（または、段階）が、予め決められたユーザ定義の制約（例えば、最大速度、最大加速度など）に基づいて計算される。そして、その制約は、動作システムの機械的な限界に対応している。これら制約、および、望ましい目標ポジションおよび速度もしくはそのいずれかを与えれば、コントローラは、望ましい動きまたは速度変化を実行するために用いられる動作プロファイルを計算する。得られる動作プロファイルは、コントローラが生成するように構成されたプロ

ファイルの形式にも依存する。その形式は、典型的には、台形またはS曲線のプロファイルである。台形プロファイルに対し、コントローラは、3つの区別された段階、加速段階、定速度段階、減速段階にしたがって動作プロファイルを計算するだろう。そのようなプロファイルは、台形の速度曲線となる。S曲線型のプロファイルは、これらの段階の移行に対応した4つの付加的な段階を加えることにより台形プロファイルを修正する。これらの付加的な段階では、定速度（もしくはゼロ速度）段階と、定加速または定減速段階と、の間を徐々に移行することができ、より滑らかな動きを与え、その動作プロファイルのより精密な制御を可能とする。

#### 【0007】

台形プロファイルは、常に、定められた最大の加速度で加速または減速を行うため、この型のプロファイルでは、S曲線プロファイルに比べて2点間の動きが速くなる傾向にある。しかしながら、台形プロファイルは、定速度（もしくはゼロ速度）段階と、加速段階と、の間の移行が急峻であり、システムに過剰なガタつきをもたらすであろう。さらに、台形の動作プロファイルを用いると、目標ポジションまたは速度に対するオーバーシュートのリスクも大きくなり精度が低下する。また、コントローラの付加的な動作、および、動作機構を望ましい目標に戻す時間が生じる。一方、S曲線プロファイルでは、定速度フェーズと、加速または減速フェーズと、の間で高精度の制御が可能となるが、2点間移動そのものに付加的な時間が生じる。

#### 【0008】

上記の説明は、単に、通常の動作制御システムが当面するいくつかの課題の概要を提供することを意図するものである。通常のシステムにおける他の課題、および、ここに説明する様々な非限定的な実施形態の対比される利点は、以下の説明により明らかとなるであろう。

#### 【発明の概要】

#### 【課題を解決するための手段】

#### 【0009】

1以上の実施形態を基本的に理解するために、以下に概要を提示する。この概要は、すべての考慮された実施形態の全容を表すものではなく、全ての実施形態の鍵となる、または、不可欠な要素を認識させるものではなく、任意の、もしくは、全ての実施形態を説明するものでもない。その目的は、後に提示されるより詳細な説明の前奏として、簡略化された形態の1以上の実施形態の概要を提示することである。

#### 【0010】

この開示の1以上の実施形態は、制御される機械システムの慣性および摩擦係数を自動的に推定するシステムおよび方法に関する。この目的を達成するために、慣性推定システムは、コントローラがモータにトルク制御信号を送るように命令することができ、トルク制御信号は、定められた最大トルク値と最小トルク値との間で時間に対し連続的に変化する。このトルク制御信号は、慣性推定システムにおいて定められたテストシーケンスに基づいて制御される。非限定的な例として、テストシーケンスは、トルク制御信号が定められた増加レートで徐々に増加し、モータを加速させることを特定できる。定められたトリガーに応答して、トルク制御信号は、ゼロまで徐々に減少し、モータを停止状態に減速させることができる。

#### 【0011】

これらの加速および減速フェーズの間、慣性推定システムは、トルク制御信号に応答するモータの速度を時間に対して測定し記録する。そして、慣性演算部は、時間的に変化するトルク信号および測定された速度曲線に基づいて、機械システムの推定慣性および推定摩擦係数のいずれか、または、両方を決定する。推定慣性および推定摩擦係数、もしくは、そのいずれかは、システムの適切な制御利得の同定を容易にするため、後に使用される。

#### 【0012】

この開示の他の形態は、制約条件に基づく時間的に最適化された動作プロファイルを効

10

20

30

40

50

率てきに生成するシステムおよび方法に係る。この目的を達成するために、コントローラの内部に配置されたプロファイル生成部は、制約条件に基づく、時間的に最適化された2点間の移動をリアルタイムに解析し、その解に基づいた軌跡を計算するために数学的なアルゴリズムを活用する。スムーズで正確な2点間の移動を得るために、プロファイル生成部は、プロファイルの加速セグメントおよび減速セグメントの少なくとも1つにおいて、時間に対し連続なジャーク基準を有するプロファイルを生成し、ST曲線タイプのプロファイルに基づいた軌跡を計算する。時間的に変化するジャーク基準を含む動作プロファイルの計算により、ここに開示されるプロファイル生成部は、通常の台形もしくはS曲線プロファイルよりもスムーズで安定した動作を創り出すことができる。

【0013】

10

プロファイル生成部により生成されたST曲線のプロファイルは、非対称な加速および減速フェーズをサポートする。従来は、非対称な加速度および減速度は、台形プロファイルによってのみサポートされており、よりスムーズなS曲線プロファイルではサポートされていない。ここに説明される技術にしたがって生成されるST曲線は、非対称な加速度および減速度に対しよりスムーズな動作プロファイルの使用を可能とする。いくつかの実施形態では、ここに説明されるプロファイル生成部は、非対称の加速度および減速度をサポートするS曲線プロファイルも生成できる。

【0014】

他の態様では、ここに説明されるプロファイル生成部の1以上の実施形態は、最終的な軌跡で使用されない軌跡のセグメントに対して計算を省略することによりその効率を改善することができる。すなわち、1以上のセグメントが使用されないケースにおいて、プロファイルの7つの全ての段階に対しプロファイルデータを計算するよりも、ここに説明されるプロファイル生成部は、与えられた軌跡に対し、最終的な動作プロファイルで使用する段階についてのみ計算を行い、コントローラの演算負荷を減少させることができる。プロファイル生成部は、与えられた2点間の移動に対し、動作プロファイルのどのセグメントがスキップされるかを自動的に決定し、その結果、残りのセグメントを計算することができる。

20

【0015】

他の態様によれば、ここに説明されるプロファイル生成部の1以上の実施形態は、プロファイルの全時間が動作コントローラのサンプリング時間の倍数となるようにすることで2点間の移動の精度および効率を向上させる。例示的な技術では、プロファイル生成部は、与えられた2点間の移動に対し、時間的に最適化された解を計算し、得られたプロファイルのそれぞれのセグメントの継続時間を決定し、これらの継続時間がサンプリング時間の倍数となるように丸めることができる。プロファイル生成部は、そのプロファイルのジャーク、加速度/減速度、速度、およびポジションを、これらの丸められた時間に適合するように再計算することができる。結果として、出力される軌跡はサンプル点に一致し、プロファイルの全時間が2つのサンプル時間の間に位置する場合に生じる小さな違いを補償する必要性が緩和される。

30

【0016】

以下の説明および添付された図面は、1以上の実施形態のいくつかの図示できる詳細な態様を説明する。しかしながら、これらの態様は、様々な実施形態の原理を用いることが可能な多くの方法のいくつかを示しており、説明される実施形態は、そのような態様の全て、および、それと同等のものを含む。

40

【図面の簡単な説明】

【0017】

【図1】簡略化された閉ループ動作制御の構成を表すブロック図である。

【図2】非限定的な慣性推定システムを例示するブロック図である。

【図3】慣性推定器に関連する入出力を表すブロック図である。

【図4】慣性推定器と、動作制御システムと、の間の相互作用を表すブロック図である。

【図5】トルクコマンド $u(t)$ と、対応するフィードバック速度 $v(t)$ と、を時間に

50

対して例示するグラフである。

【図 6】慣性部と摩擦係数部とを有する慣性推定器を表すブロック図である。

【図 7】慣性推定器が動作コントローラと比較される独立要素として機能する構成を例示するブロック図である。

【図 8】慣性推定器によって生成された推定慣性および摩擦係数を用いる動作制御チューニングの応用例を表すブロック図である。

【図 9】制御された機械システムに対する慣性および摩擦係数を推定するための手順を例示するフローチャートである。

【図 10】慣性および摩擦係数を推定するために、動作制御システムのテストシーケンスを実行する手順を例示するフローチャートである。

【図 11】動作制御システムにおける動作プロファイルの生成が可能な動作プロファイル生成システムを例示するブロック図である。

【図 12】プロファイル生成器を用いる動作コントローラを例示するブロック図である。

【図 13】例示されたポジションプロファイル生成器の入出力を表すブロック図である。

【図 14】例示された速度プロファイル生成器の入出力を表すブロック図である。

【図 15】通常の台形および S 曲線プロファイルと、例示された S T プロファイルを比較するグラフである。

【図 16】すべての 7 つの段階を用いた S 曲線プロファイルを例示するグラフである。

【図 17】定速度段階を省いた S 曲線の線動作プロファイルを例示するグラフである。

【図 18】定加速および定減速段階を省いた S 曲線プロファイルを例示するグラフである

【図 19】定加速、定速度および定減速段階を省いた S 曲線プロファイルを例示するグラフである。

【図 20】動作制御システムにおける 2 点間移動のための動作プロファイルを計算する手順を例示するフローチャートである。

【図 21】動作コントローラのサンプリング時間にしたがった制約に基づく時間的な最適動作プロファイルの計算の手順を例示するフローチャートである。

【図 22】セグメントを省略して 2 点間移動を行うための動作プロファイルを計算する手順を例示するフローチャートである。

【図 23】ここに説明する 1 以上の実施形態を実施するためのネットワーク化もしくは分散化された演算環境を例示するブロック図である。

【図 24】ここに説明する 1 以上の実施形態を実施するための演算システムもしくは動作環境を例示するブロック図である。

【発明を実施するための形態】

【0018】

図を参照しながら、いくつかの実施形態を説明する。図中の類似する符号は、類似する要素を表す。以下の説明における多数の具体的な詳細は、説明を目的とし、この開示を理解するために提供される。しかしながら、実施形態は、これらの具体的な詳細なしに、もしくは、他の方法、要素、材料などを用いて実施可能である。他の例として、構造および装置は、1 以上の実施形態の説明を容易にするためにブロック図の形態で示される。

【0019】

ここで説明されるシステムおよび方法は、制御された機械システムのための慣性および摩擦係数の推定値を生成する技術に関する。ここに開示される 1 以上の実施形態では、以下に詳細に明らかにされるテストシーケンスを通して機械システムを動作させることにより、これらのパラメータを実質的に自動的に推定することができる。このテストシーケンスの結果は、そのシステムに対して推定される正確な慣性および摩擦係数を生成するために用いることができる。これら推定されたパラメータは、実質的に、簡略化された正確なチューニングおよび動作システムの制御を容易にするために用いることができる。

【0020】

( 慣性および摩擦係数の推定 )

10

20

30

40

50



図 1 は、簡略化された閉ループ動作制御の構造を表している。コントローラ 102 は、機械的負荷 106 を動かすモータ 104 を制御するようプログラムされている。コントローラ 102、モータ 104 および負荷 106 は、例示される動作制御システムの基本的な要素を含む。非限定的に例示される用途では、負荷 106 は、単軸もしくは多軸ロボットの 1 つの軸、または、ポジショニングシステムを表す。そのような用途では、コントローラ 102 は、負荷 106 を望ましいポジションに望ましいスピードで移動させることをモータ 104 に命令する制御信号 108 を送る。制御信号 108 は、モータ 104 に直接与えても良いし、モータ 104 に供給する電力（および、モータのスピードと方向）を制御するモータ駆動部（図示しない）に与えても良い。フィードバック信号 110 は、モータ 104 または負荷 106、もしくは、その両者の現状態（すなわち、ポジション、速度など）を実質的にリアルタイムで示す。サーボ駆動システムでは、例えば、モータの絶対的または相対的なポジションを追跡する符号化器または解析器（図示しない）によりフィードバック信号 110 を生成できる。速度センサを欠くセンサ無しのシステムでは、フィードバック信号は、スピードまたはポジション推定器により提供される。移動動作の間、コントローラは、ロード 106 が正確に目標のポジションに到達するようにフィードバック信号 110 を監視する。コントローラ 102 は、フィードバック信号 110 により示される現実のポジションを目標のポジションと比較し、現実および目標のポジション間の誤差を減少または無くすように制御信号 108 を調整する。

10

#### 【0021】

他の例示的な用途では、負荷 106 は、モータ 104 で駆動される回転負荷（例えば、ポンプ、洗浄機、遠心分離器など）を表し、コントローラ 102 は、負荷の回転速度を制御する。この例では、コントローラ 102 は、モータ 104 に（制御信号 108 を介して）第 1 速度から第 2 速度へ移行する命令を与え、フィードバック信号 110 に基づいて制御信号 108 に必要な調整を行う。本用途のパラメータの推定技術は、上記の動作制御システムの例示された形式を用いることに限定されるのではなく、いかなる動作制御の用途にも好適に応用できることが認識されるべきである。

20

#### 【0022】

望ましいポジションまたは速度と、目標のポジションまたは速度と、の間の誤差（フィードバック信号 110 により伝えられる）に応答してコントローラ 102 によって生成される制御信号出力は、制御ループの利得係数に依存する。好適な利得の選定は、制御を受ける機械システムの物理的な特性に依存するため、設計技術者は、しばしば試行錯誤的な手法（すなわち、制御ループのチューニング）を用いて好適な利得係数を明らかにしなければならない。例えば、大きな慣性（加速または減速に対する抵抗）を有する機械システム、特に、目標のポジションまたは速度への速い収束が求められる用途では、新しいポジションまたは速度へ動き出すために比較的大きな初期トルクを必要とするだろう。しかしながら、大きなトルクは、オーバーシュートの可能性を大きくし、システムを目標に戻す逆補正を必要とする。最適な利得設定をしない場合、目標のポジションまたは速度に落ち着く前に、システムは補正のための相互作用を実施するため、望まない機械振動が生じる。そのような振動は、システムを不安定とし、システム遅延を生じさせる。そして、システムを安定させるために求められる付加的な動作の結果として過剰な電力を消費する。機械システムが与えられる制御信号に応答する程度にモータの摩擦も影響し、制御システムをチューニングする場合に考慮すべき因子となる。

30

40

#### 【0023】

機械システムの慣性および摩擦係数の正確な推定値を知れば、制御システムのチューニングは、簡略化できる。これらのパラメータの知識は、動作中のシステムの性能も向上させる。これに応じて、本用途の 1 以上の実施形態では、制御された機械システムの慣性および摩擦係数を、実質的に自動化された態様で正確に推定することができる。

#### 【0024】

図 2 は、機械システムの慣性および摩擦係数の推定値を生成できる非限定的な慣性の推定システムを例示するブロック図である。慣性推定器 202 は、トルクコマンド生成部 2

50

04、速度監視部206、慣性演算部208、摩擦係数演算部210、インターフェース部212、1以上のプロセッサ214およびメモリ216を含む。多様な実施形態において、トルクコマンド生成部204、速度監視部206、慣性演算部208、摩擦係数演算部210、インターフェース部212、1以上のプロセッサ214およびメモリ218のうちの1つ以上は、慣性推定器202の1以上の機能を実行するために、相互に電氣的もしくは通信可能に結合でき、また、相互に電氣的および通信可能に結合できる。

#### 【0025】

いくつかの実施形態では、トルクコマンド生成部204、速度監視部206、慣性演算部208、摩擦係数演算部210およびインターフェース部212は、メモリ216に記憶され、プロセッサ214により実行されるソフトウェアの命令を含む。慣性推定器202は、図2に表されない他のハードウェアおよびソフトウェア部品とも相互に作用できる。例えば、プロセッサ214は、キーボード、マウス、表示モニター、タッチスクリーン、もしくは、他のインターフェース器機など、1以上の外部ユーザとのインターフェース装置と相互作用する。

10

#### 【0026】

インターフェース部212は、ユーザの入力を受け入れ、任意の好適な形式（例えば、視覚、音声、触覚など）によりユーザに出力を提供する。ユーザの入力は、例えば、慣性推定シーケンス（以下に詳細に説明される）を実行する際に慣性推定器で使用されるユーザ入力のパラメータでも良い。トルクコマンド生成部204は、定義されたテストシーケンスにしたがって、時間的に連続して変化するトルク制御命令を出力するように構成できる。速度監視部206は、機械システムの速度データを慣性および摩擦係数の計算に使用するために受信することができる。いくつかの実施形態では、速度監視部206は、トルクコマンド生成部204によって生成されたトルク制御命令が適用され、それに応答したモータの速度を測定し記録する。また、速度監視部206は、別の測定器具から測定された速度データを受信することもできる。慣性演算部208および摩擦係数演算部210は、トルクコマンド生成部204により生成された時間的に変化するトルクコマンド、および、速度監視部206により取得された速度曲線に基づいて、慣性および摩擦係数をそれぞれ計算するように構成される。1以上のプロセッサ214は、開示されたシステムおよび方法、もしくは、そのいずれかを参照して、以下に説明される1つ以上の機能を実施する。メモリ216は、コンピュータが読み取り可能な記憶媒体であれば良く、開示されたシステムおよび方法、もしくは、そのいずれかを参照して、以下に説明される機能を実施するために、コンピュータが実行可能な命令および情報、もしくはそのいずれかを記憶する。

20

30

#### 【0027】

慣性推定器は、テストシーケンスを通してシステムを動作させ、その結果に基づいて推定値を計算することにより機械システムの慣性および摩擦係数に対する推定値を生成する。図3は、慣性推定器302（慣性推定器202に類似）に関連する入出力を表すブロック図である。慣性推定器302は、動作システムを駆動するモータに、与えられたトルクで特定された方向に回転することを命令するトルクコマンド310を生成することができる。一定のトルク値の間の遷移を突発的なステップで生じさせ、結果としてステップ状のトルクを出力させる1つ以上の一定のトルクコマンドを発するよりも、慣性推定器302は、最大および最小のトルク値の間でトルク値が時間に対して連続的に変化するようトルクコマンド310を制御する。以下により詳細に議論するように、ユーザが定義したパラメータを有するテストシーケンスにしたがって、慣性推定部302は、トルクコマンド310を介して出力されるトルク値を制御する。

40

#### 【0028】

動作システムは、慣性推定器から出されたトルクコマンド310にしたがって加速または減速を行い、動作制御システムからの速度帰還304が慣性推定器302に与えられる。速度帰還304は、動作システムの速度を時間に対して表す。例示されたテストシーケンスでは、慣性推定器302は、速度帰還304およびユーザにより定義された1以上の

50

設定点の関数としてトルクコマンド 3 1 0 を制御することができる。ユーザ定義の設定点は、トルクコマンド信号の上限および下限を決める 1 以上の限界トルク 3 0 6、および、トリガー速度値を決める 1 以上の速度検出点 3 0 8、または、そのいずれかを含む。トリガー速度値は、トルクコマンド 3 1 0 を制御するため、および、推定値を生成するために使用される。

#### 【 0 0 2 9 】

テストシーケンスを完了すると、慣性推定器 3 0 2 は、動作システムの慣性の推定値 3 1 2 および動作システムの摩擦係数の推定値 3 1 4 を生成する。慣性推定器 3 0 2 は、動作システムに出力され、速度帰還 3 0 4 に対応するトルクコマンド 3 1 0 に基づいてこれらの推定値を決定する。1 以上の実施形態において、慣性推定器 3 0 2 は、トルク曲線（トルクコマンド 3 1 0 に対応）および速度曲線（速度帰還 3 0 4 に対応）の選択された部分を時間に対して積分し、これら積分の関数として推定慣性 3 1 2 および推定摩擦係数 3 1 4 を計算することができる。

#### 【 0 0 3 0 】

図 4 は、例示されたテストシーケンスを実行する間の慣性推定器と動作制御システムとの相互作用を表している。この例では、動作システム 4 2 4 は、コントローラ 4 1 8 により与えられる制御信号 4 2 0 に応答するモータ 4 2 4 を含む。モータ 4 2 4 は、ポジション軸、機械の回転部または他のモータ駆動の負荷のような負荷（図示しない）を駆動するために用いられる。コントローラ 4 1 8 は、モータ 4 2 4 のリアルタイムの状態を示すデータ（例えば、ポジション、スピードなど）を与えるフィードバック 4 2 2 を監視する。

#### 【 0 0 3 1 】

図示した例では、慣性推定器 4 0 2 は、コントローラ 4 1 8 から分離した要素として明示されている。このような構成では、慣性推定器 4 0 2 は、コントローラ 4 1 8 または動作システム 4 2 4 の他の要素と、任意の好適な通信手段を介してデータをやりとりすることができる。通信手段は、有線または無線のネットワーク、配線されたデータリンクもしくは他の通信手段を含むが、これらに限定される訳では無い。他の実施形態では、慣性推定器 4 0 2 は、コントローラ 4 1 8 に統合されても良い。例えば、慣性推定器 4 0 2 は、コントローラのオペレーティングシステムの機能部、および、コントローラ 4 1 8 上に存在する 1 以上のプロセッサにより実行される制御ソフトウェア、もしくは、そのいずれかであっても良い。また、慣性推定器 4 0 2 は、コントローラ 4 1 8 の中で他の機能要素とデータをやり取りする回路基板または集積回路のようなハードウェア部品であっても良い。慣性推定器 4 0 2 の他の好適な実施は、本開示のいくつかの実施形態の範疇にある。

#### 【 0 0 3 2 】

試験の前に、1 以上のユーザ定義のパラメータ 4 1 2 が、インターフェース部 4 0 6（図 2 に関連して説明したインターフェース部 2 1 2 に類似）を介して慣性推定器 4 0 2 に与えられる。これらのパラメータは、トルクコマンド生成部 4 0 8（図 2 のトルクコマンド性西部 2 0 4 に類似）により生成されるトルクコマンドの上限および下限を決める最大トルク  $u_{max}$  および最小トルク  $u_{min}$  を含む。いくつかの実施形態では、慣性推定器 4 0 2 は、ユーザにより決定される最大トルク  $u_{max}$  のみを必要とし、決定された最大トルクの大きさを進行方向および逆方向の両方に対する限界値として用いる。他の実施形態では、慣性推定器 4 0 2 は、 $u_{max}$  および  $u_{min}$  の両方の値を受け入れ、進行方向および逆方向のそれぞれに対し、異なるトルク設定点を許容する。 $u_{max}$  および  $u_{min}$  に対して選択された値は、動作システム 4 2 4 の予定された動作限界に対応し、これにより、システムのトルクプロファイル全体に対する動作システム 4 2 4 の特性に基づいて慣性および摩擦係数を決定することが許容される。以下により詳細に説明するように、ユーザ定義のパラメータ 4 1 2 は、テストシーケンスの段階を定義するために用いられる臨界速度を決める 1 以上の速度検出点（ $v_1$ 、 $v_2$ 、 $v_3$ 、・・・）も含む。

#### 【 0 0 3 3 】

インターフェース部は、ユーザ定義のパラメータ 4 1 2 をトルクコマンド生成部 4 0 8 に与える。試験を開始したとき、トルクコマンド生成部 4 0 8 は、動作システム 4 2 4 に

トルクコマンド 414 を出力する。トルクコマンド生成部 408 は、時間に対して連続的にトルクコマンドを変化させるため、トルクコマンド 414 は、 $u(t)$  のように表される。図 4 に表された構成では、慣性推定器 402 は、トルクコマンド 414 をコントローラ 418 に送り、コントローラ 418 は、モータ 424 に指示された方向に、指示されたトルクで回転するように（制御信号 420 を介して）命令する。モータの回転にしたがい、速度監視部 410 は、コントローラ 418 から速度データ 416 を読み取る。（コントローラ 418 自身は、フィードバック 422 を介してモータ 424 の速度を計測する。）時間に対して測定された速度 416 は、 $v(t)$  のように表される。

#### 【0034】

試験の進行にしたがい、トルクコマンド生成部 408 は、予め決定されたテストシーケンスにしたがってトルクコマンド 414 を変化させ、テストシーケンスの各フェーズは、ユーザ定義のパラメータ 412 に比較された速度帰還 416 により開始される。テストシーケンスの例は、トルクコマンド  $u(t)$  の例、および、対応する速度帰還  $v(t)$  を時間に対するグラフで示した図 5 を参照して以下に説明される。トルクのグラフ 502 に示すように、トルクコマンド信号  $u(t)$  は、 $u_{max}$  および  $u_{min}$  により拘束される。速度グラフ 504 に示された速度監視点  $v_1$ 、 $v_2$  および  $v_3$  は、テストシーケンスのフェーズの移行を決定する。 $u_{max}$ 、 $u_{min}$ 、 $v_1$ 、 $v_2$  および  $v_3$  の値は、（例えば、図 4 のユーザ定義のパラメータ 412 のように）試験の前にユーザにより決定される。

#### 【0035】

時間  $t = 0$ （ゼロ）において試験を開始したとき、適用されるトルク信号  $u(t)$  およびモータ速度  $v(t)$  は、共にゼロである。最初に、トルクコマンド生成部は、負のトルク信号を動作システムに送り、動作システムを負の方向に加速する。この試験の第 1 フェーズでは、トルクコマンド生成部は、モータ速度  $v(t)$  が  $v_1$  に達するか、トルクコマンド  $u(t)$  が  $u_{min}$  に達するまで、トルクコマンド  $u(t)$  を徐々に減少させる。この例では、モータ速度  $v(t)$  は、時間  $t = t_1$  において  $v_1$  に達し、試験の第 2 フェーズを開始させる。グラフ 502 に示すように、トルクコマンド  $u(t)$  は、時間  $t = 0$  と  $t = t_1$  との間において、実質的に一定のレートで連続的に減少する。1 以上の実施形態において、トルクコマンドが減少または増加するレート（すなわち、 $u(t)$  の傾き）は、（例えば、インターフェース部 406 を介した）慣性推定器 402 のユーザ定義のパラメータとして構成できる。

#### 【0036】

試験の第 2 フェーズ（時間  $t = t_1$  において開始）に対して、トルクコマンド生成部は、モータ速度  $v(t)$  が速度検出点  $v_3$  に達するか、トルクコマンド  $u(t)$  がトルク設定点  $u_{max}$  に達するまで、トルクコマンド  $u(t)$  を徐々に増加させる。この例では、トルクコマンド  $u(t)$  は、モータ速度  $v(t)$  が速度検出点  $v_3$  に達する前に、上限  $u_{max}$  に達している。モータは、この時点でまだ加速しているため、トルクコマンド生成部は、速度  $v(t)$  が  $v_3$  に達するまで、トルクコマンド信号を  $u_{max}$  に維持する。速度グラフ 504 に表されるように、モータ速度は、時間  $t = t_4$  において  $v_3$  に達する。速度  $v(t)$  が、トルクコマンド信号が  $u_{max}$  に達した後、定義された時間までに速度検出点  $v_3$  に達しなければ（すなわち、速度検出点  $v_3$  が何かの理由で動作システムの物理的な限界速度よりも高く設定されたとすれば）、慣性推定器は、好適なタイムアウト処理ルーティンを始めることができる。タイムアウト処理ルーティンは、例えば、テストシーケンスを中断し、インターフェース部 406 を介してエラーメッセージを表示することを含む。

#### 【0037】

このフェーズの間、動作システムは  $v_3$  に向けて加速しているので、速度は、テストシーケンスの加速フェーズを示す速度検出点  $v_2$  を通過する。以下により詳細に議論するように、速度検出点  $v_2$  は、ゼロより大きく、 $v_3$  より小さく設定され、テストシーケンスの加速フェーズの開始と、減速フェーズの終了と、を表すために用いられる。

#### 【0038】

10

20

30

40

50

モータ速度が  $v_3$  に達したことを検出すれば、トルクコマンド生成部は、時間  $t = t_4$  においてトルクコマンド  $u(t)$  を徐々に減少させることにより、試験の第3フェーズを開始する。トルクコマンド  $u(t)$  が減少するにしたがい、モータは、トルクコマンド  $u(t)$  の値が動作システムの摩擦力よりも小さくなるまで、しばらくの間、加速を続ける。そして、トルクコマンド  $u(t)$  の値が動作システムの摩擦力よりも小さくなった時、モータは減速を開始する。モータは、時間  $t = t_3$  において速度が  $v_3$  に達した時、まだ加速しているため、トルクコマンドが減少を始めた後も、いくらかの時間の間、 $v_3$  を越えた状態を続ける。テストシーケンスの定義にしたがって、トルクコマンド生成部は、モータ速度  $v(t)$  が速度検出点  $v_3$  に戻る（時間  $t = t_6$ ）まで、 $u(t)$  を減少させ続ける。その後、モータ速度  $v(t)$  が速度検出点  $v_2$  に戻る（時間  $t = t_7$ ）まで、 $u(t)$  を一定に保持する。この時点で、慣性推定器は、機械システムの慣性および摩擦係数の推定を計算するために必要なデータを取得する。トルクコマンド生成器は、トルクコマンド信号  $u(t)$  をゼロに戻し（時間  $t = t_8$ ）、グラフ504に表すように、 $v(t)$  曲線の端を漸減させることにより、動作システムを惰性で停止させる。

【0039】

図5に関連して説明した上記のテストシーケンスは、一例を表すことを意図するだけであり、テストシーケンスを限定するものではない。時間に対してトルクコマンド  $u(t)$  を連続的に変化させ、動作システムの対応する速度プロファイル  $v(t)$  を測定する任意の好適なテストシーケンスは、ここに開示されたいくつかの実施形態の範疇にあることが理解されるべきである。例えば、前述した例では、それぞれの速度検出点に到達した速度  $v(t)$  に応答して方向を変化させるトルクコマンド  $u(t)$  を説明したが、いくつかのテストシーケンスでは、トルクコマンドの方向は変化させずに、速度検出点に到達した時、トルクコマンド  $u(t)$  の増加もしくは減少のレートを変えるだけのフェーズを含んでも良い（すなわち、増加するトルクコマンドは、フェーズ検出点に到達した  $v(t)$  に応答し、よりゆっくりしたレートで増加を続けても良い）。

【0040】

前述したテストシーケンスを実施する場合、慣性推定器402は、トルクコマンド生成部408により生成されたトルクコマンド信号  $u(t)$  と、速度監視部410により読み取られた対応するモータ速度  $v(t)$  の両方を記録する。これらのトルク曲線および速度曲線は、慣性および摩擦係数の正確な推定値をその曲線に基づいて計算できるように、動作システム424を特徴付ける。1以上の実施形態において、慣性推定器は、 $u(t)$  および  $v(t)$  の積分に基づいてこれらの推定値を計算する。次に、動作システムに対する慣性および摩擦係数の推定値を導き出すために  $u(t)$  および  $v(t)$  の積分を活用する非限定的な技術を例示する。

【0041】

動作システムは、微分方程式で表すことができる。

【数1】

$$J\dot{v}(t) = -Bv(t) + u(t) \quad (1)$$

ここで、 $J$  は慣性であり、 $B$  は摩擦係数である。 $u(t)$  はトルクコマンド信号、 $v(t)$  は、トルク信号  $u(t)$  に応答する動作システムの対応する速度である（例えば、 $u(t)$  および  $v(t)$  は、図4および図5に関連して説明されている）。

【0042】

式(1)の両側を、それぞれ加速および減速段階において積分すると次式を得る。

【数2】

$$J\Delta v_{acc}(t) = -B \int v_{acc}(t) + \int u_{acc}(t) \quad (2)$$

10

20

30

40

【数 3】

$$J\Delta v_{dec}(t) = -B \int v_{dec}(t) + \int u_{dec}(t) \quad (3)$$

ここで、 $u_{acc}(t)$  および  $v_{acc}(t)$  は、それぞれ  $u(t)$  および  $v(t)$  の一部であり、テストシーケンスの加速フェーズに対応する。 $u_{dec}(t)$  および  $v_{dec}(t)$  は、それぞれ  $u(t)$  および  $v(t)$  の一部であり、テストシーケンスの減速フェーズに対応する。

【0043】

式(2)および(3)を解いて、慣性  $J$  および摩擦係数  $B$  を求めることができる。

10

【数 4】

$$J = \frac{\int u_{dec}(t) \int v_{acc}(t) - \int u_{acc}(t) \int v_{dec}(t)}{\Delta v_{dec}(t) \int v_{acc}(t) - \Delta v_{acc}(t) \int v_{dec}(t)} \quad (4)$$

【数 5】

$$B = \frac{\Delta v_{dec}(t) \int u_{acc}(t) - \Delta v_{acc}(t) \int u_{dec}(t)}{\Delta v_{dec}(t) \int v_{acc}(t) - \Delta v_{acc}(t) \int v_{dec}(t)} \quad (5)$$

20

【0044】

図4に表されたトルク曲線および速度曲線の例では、加速フェーズは、速度  $v(t)$  が最初に速度検出点  $v_2$  に達した時(時間  $t = t_3$ )に始まり、トルク信号  $u(t)$  がゼロをよぎった時(時間  $t = t_5$ )に終わる。この加速フェーズの終点における動作システムの速度は、(グラフ504に示される)  $v_4$  として記録される。減速フェーズは、トルク信号  $u(t)$  がゼロをよぎった時(時間  $t = t_5$ )に始まり、速度  $v(t)$  が  $v_2$  に戻った時(時間  $t = t_7$ )に終わる。慣性推定器402は、式(4)および(5)に基づいて推定慣性および推定摩擦係数を導出するために、これらの加速および減速フェーズの描写を認識するように構成できる。加速フェーズおよび減速フェーズを描写するための他の基準も、ここに開示されたいくつかの実施形態の範囲にあることを認識すべきである。

30

【0045】

これらの加速フェーズおよび減速フェーズの定義が与えられると、 $u_{acc}(t)$  および  $u_{dec}(t)$  の積分範囲は、それぞれ  $U_{acc}$  および  $U_{dec}$  と表示されたグラフ502の網掛け領域で表され、 $v_{acc}(t)$  および  $v_{dec}(t)$  の積分範囲は、それぞれ  $V_{acc}$  および  $V_{dec}$  と表示されたグラフ504の網掛け領域で表される。したがって、 $U_{acc}$ 、 $U_{dec}$ 、 $V_{acc}$  および  $V_{dec}$  は、次のように決定される。

【数 6】

$$U_{acc} = \int u_{acc}(t) \quad (6)$$

40

【数 7】

$$U_{dec} = \int u_{dec}(t) \quad (7)$$

【数 8】

$$V_{acc} = \int v_{acc}(t) \quad (8)$$

50

【数 9】

$$V_{dec} = \int v_{dec}(t) \quad (9)$$

【0046】

式(6)～(9)を式(4)～(5)に代入すると、慣性 J および摩擦係数 B は、次のように表せる。

【数 12】

$$J = \frac{U_{dec} V_{acc} - U_{acc} V_{dec}}{\Delta v_{dec}(t) V_{acc} - \Delta v_{acc}(t) V_{dec}} \quad (12) \quad 10$$

【数 13】

$$B = \frac{\Delta v_{dec}(t) U_{acc} - \Delta v_{acc}(t) U_{dec}}{\Delta v_{dec}(t) V_{acc} - \Delta v_{acc}(t) V_{dec}} \quad (13)$$

ここで、速度変化  $v_{acc}(t)$  および  $v_{dec}$  は、次のように定義される。

【数 10】

$$\Delta v_{acc}(t) = v_4 - v_2 \quad (10) \quad 20$$

【数 11】

$$\Delta v_{dec}(t) = v_2 - v_4 \quad (11)$$

【0047】

式(12)および(13)は例示であり、連続的なトルクおよび速度データに基づいて動作システムの推定慣性および推定摩擦係数を計算するための非限定的な式である。連続的なトルク信号および対応する速度曲線の積分を介してこれらのパラメータを計算するための任意の好適な式は、ここに開示するいくつかの実施形態の範囲内であることが認識されるべきである。

30

【0048】

図4および図5に関連して上記のテストシーケンスを完了した場合、慣性推定器は、式(12)および(13)(または他の好適な式)を、慣性および摩擦係数の推定量を導出するための試験により得られた連続的なトルクデータ  $u(t)$  およびモータ速度データ  $v(t)$  に適用することができる。図6は、ここに開示する1以上の実施形態にしたがった、慣性演算部606と、摩擦係数演算部608と、を有する慣性推定器602を表すブロック図である。慣性推定部602は、慣性演算部606および摩擦係数演算部608の両方を含むように表されているが、慣性推定部602のいくつかの実施形態は、この開示の範囲を逸脱することなく、これらの演算部の1つだけを含むことができる。すなわち、慣性推定器602は、慣性および摩擦係数のいずれか1つ、または、その両方を計算するように構成しても良い。

40

【0049】

トルクデータ  $u(t)$  および速度データ  $v(t)$  が得られた後、トルクコマンド生成部604(トルクコマンド生成部408および204に類似)は、慣性演算部606および摩擦係数演算部608(図2の慣性演算部208および摩擦係数演算部210にそれぞれ類似)にトルクデータを与える。同様に、速度監視部606は、慣性演算部606および摩擦係数演算部608に得られた速度データ  $v(t)$  を与える。1以上の実施形態によれば、慣性推定器602は、上記の式(6)～(9)を用いて  $U_{acc}$ 、 $U_{dec}$ 、 $V_{acc}$  および  $V_{dec}$  の値を導出できるように、トルクデータおよび速度データを加速フェー

50

ズのデータ ( $u_{acc}(t)$  および  $v_{acc}(t)$ ) と、減速フェーズのデータ ( $u_{dec}(t)$  および  $v_{dec}(t)$ ) と、に分離する。

#### 【0050】

慣性演算部は、 $u_{acc}(t)$ 、 $u_{dec}(t)$ 、 $v_{acc}(t)$  および  $v_{dec}(t)$  を積分し、(例えば、式(12)もしくはその変形に基づいて)積分値の関数として推定慣性  $J_{610}$  を計算する。同じように、摩擦係数演算部 608 は、(例えば、式(13)に基づいて)積分値の関数として推定摩擦係数  $B_{616}$  を計算することができる。その結果、慣性推定器が動作する個別の用途の要求にしたがって、慣性推定器 602 は、推定慣性  $J_{610}$  と推定摩擦係数  $B_{612}$  とを出力することができる。例えば、慣性推定器 602 は、慣性  $J_{610}$  と摩擦係数  $B_{612}$  とを動作制御部 614 に与え、動作制御部 614 は、1以上の利得係数のチューニングを容易にするために  $J$  および  $B$  の値を用いる。慣性推定器 602 は、 $J$  および  $B$  の推定値を見て別の動作制御もしくはチューニングの用途にマニュアルで入力できるように、(例えば、インターフェース部 212 を介して)表示装置に出力することもできる。動作システムの慣性  $J_{610}$  および摩擦係数  $B_{612}$  の正確な推定は、チューニングプロセスを簡略化し、正確なパラメータチューニングを容易にする。この結果、緻密でエネルギー効率の良い機械動作が得られる。さらに、慣性推定器は、(1以上の一定のトルクコマンドに対するシステムの応答に基づいて推定するのではなく)動作システムのトルクプロファイル全体から収集されたデータに基づいて  $J$  および  $B$  の値を計算するので、慣性推定器により導出される慣性および摩擦係数の推定は、動作システムの動作範囲全体において、より正確である。

10

20

#### 【0051】

前述した例では、慣性推定器は、トルクコマンド  $u(t)$  を送り、動作制御部(例えば、図4のコントローラ 418)を介して速度帰還  $v(t)$  を受け取り、コントローラを介して動作する分散した部品、もしくは、コントローラとして集積化された部品のいずれかであるように説明したが、他の構成も、ここに開示するいくつかの実施形態の範囲に含まれる。例えば、図7は、慣性推定器 706 がコントローラ 702 から独立した要素として動作する構成を表している。この例では、慣性推定器 706 は、コントローラ 702 とは独立した自分のトルクコマンド信号を生成することができる。試験され、制御されるモータ 704 は、スイッチ 712 の状態に依存してコントローラ 702 または慣性推定器 706 からトルクコマンド信号を受信することができる。モータ 704 からの速度帰還 710 は、コントローラ 702 および慣性推定器 706 の両方に与えられる。テストシーケンスの間は、スイッチ 712 は、慣性推定器 706 からトルクコマンド  $u(t)$  が伝達されるようにセットされる。前述の例で説明したように試験が実行され、慣性推定器 706 は、動作システムの慣性  $J$  および摩擦係数  $B$  の推定値を生成する。そして、慣性推定器 706 は、 $J$  および  $B$  の推定値をコントローラ 702 に与え、コントローラ 706 は、好適な制御利得係数または他の制御パラメータを決定するために、その値を用いることができる。一旦、制御パラメータが設定されれば、スイッチ 712 は、コントローラ 702 からモータ 704 へトルクコマンド 708 を与える位置に切り替えられ、 $J$  および  $B$  に基づいて導出された制御利得係数を用いてモータシステムの通常動作が実行される。

30

40

#### 【0052】

図8は、慣性推定器により生成された推定慣性および推定摩擦係数を用いた動作制御チューニングの適用例を表している。この例では、チューニング部 804 がコントローラ 806 のために制御利得を調整し、コントローラ 806 は、モータ駆動の動作システム(図示しない)の動作を制御する。慣性推定器 802 は、上記の技術により動作システムの慣性  $J_{808}$  および摩擦係数  $B_{810}$  の推定値を生成することができる。具体的には、慣性推定器 802 は、コントローラ 806 に連続したトルクコマンドを動作システムのモータに送るよう命令し、トルクコマンドは、予め定められたテストシーケンスにしたがって、時間に対して連続的に変化する。また、慣性推定器 802 が(図7に表す構成例のように)コントローラ 806 とは独立に動作する実施形態では、慣性推定器 802 は、それ自身の連続したトルクコマンドを生成し、動作システムに出力しても良い。テストシーケ

50



スは、加速および減速フェーズを含むことができ、その間、慣性推定器 802 は、適用されたトルクコマンドに応答する動作システムの速度を監視および記録する。テストシーケンスの結果に対し、時間的に変化するトルクコマンド信号、および、対応して時間的に変化する動作システムの速度の積分（例えば、式（12）および（13））に基づいて、慣性推定器 802 は、慣性 J 808 および摩擦係数 B 810 の推定値を計算することができる。

#### 【0053】

慣性推定器 802 は、慣性 J 808 および摩擦係数 B 810 をチューニング部 804 に与えることができる。代わりに、慣性推定器 802 は、J および B の値をユーザインターフェースに与え、ユーザがチューニング部 804 に推定慣性と推定摩擦係数をマニュアルで入力しても良い。J および B、もしくは、そのいずれかの知識は、動作システムの機械的特性に基づいてチューニング部 804 に 1 以上の制御利得 812 の好適な推定値を生成させる。チューニング部 804 は、望ましい動作特性を得るためにインターフェース 816 を介してユーザによりマニュアルで調整された制御システムのバンド幅 814（例えば、重複した周波数）と、慣性 J および摩擦係数 B 810、もしくは、そのいずれかと、の関数として制御利得 812 の好適な値を生成できる。

10

#### 【0054】

通常の使用において、ここに説明される慣性推定器は、通常動作前の動作制御システムを最初に準備する間に、動作システムの慣性 J および摩擦係数 B の信頼できる推定値を生成することができる。具体的には、慣性推定器は、制御パラメータ（例えば、制御利得係数）の構成およびチューニングに関連して使用される。一度、設定され、後でシステムを再調整することが決定されなければ、システムが動き出した後もこれらのパラメータは、通常、固定されたままである。しかしながら、いくつかの実施形態では、慣性推定器は、動作時間の間に周期的もしくは連続的に J および B の値を自動で再計算するように構成できる。そのような構成を用いれば、J および B の推定に基づいた制御パラメータは、通常動作の間に過渡的に調整され、（例えば、機械的摩滅、モニタで見られる負荷の変化などの結果としての）段階的变化に対し、実質的にリアルタイムで、動作システムの機械的特性の補正を行うことができる。

20

#### 【0055】

図 9 および図 10 は、いくつかの開示された態様にしたがう、いくつかの手順を表している。説明の簡略化のために、手順は一連の行為として示され、説明されるが、開示された態様は、その行為の順に限定されず、いくつかの行為は、ここに示され、説明されたものと異なる順で、他の行為と同時に行われても良いし、そのいずれかであっても良いことを理解し、認識すべきである。例えば、当業者であれば、手順は、状態図のように相互に関連する状態もしくは事象の連続として選択可能に表されることを理解し、認識するであろう。さらに、いくつかの開示された態様にしたがって手順を実行するために、表示された全ての行為が求められる訳ではない。なお、この開示を通じて以下に示される手順は、そのような手順のコンピュータへの転送を容易にする製造物に記憶させることが可能であることを、さらに認識すべきである。

30

#### 【0056】

図 9 は、制御された機械システムの慣性および摩擦係数を推定するための手順例 900 を表している。ステップ 902 において、連続したトルクコマンド  $u(t)$  は、動作システムのコントローラに送られ、トルクコマンド  $u(t)$  は、決められた最大トルクおよび最小トルクの設定点の間で時間と共に変化する。1 以上の実施形態において、トルクコマンド  $u(t)$  は、出力  $u(t)$  がテストシーケンスのフェーズと、1 以上のユーザ定義の設定点と比較した機械システムの応答と、に依存するように、定められたテストシーケンスにしたがう。テストシーケンスは、モータスピードの増加および減少にそれぞれ対応する加速フェーズおよび減速フェーズの両方を備える。2 方向の試験では、トルクコマンド  $u(t)$  は、テストシーケンスの間、正または負のトルク値の間で変化し、その試験の間に、動作システムを両方向に加速する。

40

50

## 【 0 0 5 7 】

ステップ 9 0 4 では、トルクコマンド  $u(t)$  に応答した動作システムの速度  $v(t)$  が記録される。その結果、テストシーケンスの完了により、時間  $t = 0 \sim t_{end}$  に対し、適用されたトルクコマンド  $u(t)$  の曲線データ、および、結果として得られる動作システムの速度  $v(t)$  の曲線データの両方が得られる。ここで、 $t_{end}$  は、テストシーケンスの継続時間である。

## 【 0 0 5 8 】

ステップ 9 0 6 では、トルク曲線  $u(t)$  および速度曲線  $v(t)$  の積分に基づいた慣性および摩擦係数の少なくともいずれか 1 つの推定値が計算される。1 以上の実施形態では、 $u(t)$  および  $v(t)$  の曲線は、加速フェーズおよび減速フェーズに分割され、加速フェーズおよび減速フェーズのそれぞれの積分に基づいて慣性および摩擦係数を計算することができる。（例えば、式 ( 1 2 ) および ( 1 3 ) または他の好適な式を用いる。）ステップ 9 0 8 では、動作システムの 1 以上のパラメータを、ステップ 9 0 6 で計算された推定慣性および摩擦係数、または、そのいずれかの関数として設定する。非限定的な例として、ステップ 9 0 2 ~ 9 0 6 にしたがって計算された推定慣性および推定摩擦係数、もしくは、そのいずれかの基づいて 1 以上の制御利得係数を設定できる。

## 【 0 0 5 9 】

図 1 0 は、動作制御システムにおいて、慣性および摩擦係数を推定するために実行される手順例 1 0 0 0 を表している。ステップ 1 0 0 2 では、動作システムに与えられるトルクコマンドは、最大トルクの設定点に達するまで、もしくは、適用されたトルクコマンドに  
20  
応答して動作システムが第 1 の速度検出点（例えば、図 5 の速度検出点  $v_3$ ）に加速されるまで、連続的に増加する。最大トルクの設定点および第 1 の速度検出点は、動作システムの上方動作限界に対応し、（例えば、動作システムの正常動作における最大トルクおよび速度として）試験の前に設定できる。トルクコマンドの増加レートもユーザが定めることができる。1 以上の実施形態において、動作システムの速度が第 1 の速度監視点に達する前にトルクコマンドが最大トルク値に達した場合、動作システムが第 1 の速度検出点に加速されるまで、トルクコマンドは最大トルク値に保持される。動作システムの速度が定められたタイムアウト時間内に第 1 の速度検出点に達しない場合、適当なタイムアウト処理を開始する。

## 【 0 0 6 0 】

動作システムは、停止状態から第 1 の速度検出点まで加速されるので、その速度は、第 2 の速度検出点（例えば、図 2 の速度検出点  $v_2$ ）を通過するだろう。ここで、第 2 の速度検出点は、ゼロよりも大きく、第 1 の速度検出点よりも小さい。試験の加速フェーズは、速度が最初に第 2 の速度検出点に到達した時に始まる。

## 【 0 0 6 1 】

動作システムが第 1 の速度検出点に加速されたことを検出すれば、トルクコマンドは、ステップ 9 0 4 においてトルクコマンドが最初トルクの設定点に到達するまで連続的に減少される。この場合、ステップ 1 0 0 2 において第 1 の速度検出点に達した時、動作システムは加速しているので、ステップ 1 0 0 4 においてトルク信号が減少し始めてでも、しばらくの間、その速度は第 1 の速度検出点を越えて増加する。トルクコマンド信号の減少は、実質的に、動作システムを減速し、第 1 の速度検出点に戻す。ステップ 1 0 0 2 のように、トルクコマンドを減少させるレートは、ユーザ定義のパラメータとして構成できる。この例では、動作システムが第 1 の速度検出点に戻る前に、トルクコマンドはゼロに減少し、最小トルクの設定点に達するまで、もしくは、第 1 の速度検出点に減速するまで、負の方向に減少を続ける。すなわち、トルクコマンドは、テストシーケンスのこのフェーズの間にゼロ点を横切る。これは、加速フェーズの終わりと、減速フェーズの始まりを示す。ステップ 1 0 0 2 と同じように、動作システムが第 1 の速度検出点に達する前に、トルクコマンドが最小トルクの設定点まで減少すれば、トルクコマンドは、第 1 の速度検出点に達するまで、最小トルク値に保持される。

## 【 0 0 6 2 】

動作システムの速度が第 1 の速度検出点に戻った時、第 1 の速度検出点に到達した時点のトルクコマンド値は、動作システムが減速し第 2 の速度検出点に戻るまで、ステップ 1006 における一定値として維持される。これは、減速フェーズの終了のトリガとなる。

#### 【0063】

ステップ 1008 において、トルクコマンドがステップ 1004 の間にゼロをよぎる時間が検出される。この時間は、 $T_{CROSSOVER}$  として示され、テストシーケンスの加速フェーズと減速フェーズの境界を決めるために用いられる。(図 4 に関連した上記の例では、 $T_{CROSSOVER} = t_5$  である。) ステップ 1010 では、動作システムのトルクコマンド曲線および対応する速度曲線の加速フェーズの部分において積分が実行される。すなわち、トルクコマンドデータは、時間  $T_0$  から  $T_{CROSSOVER}$  まで積分 10  
される。ここで、 $T_0$  は、加速フェーズの開始時間(速度が第 2 の速度検出点に最初に交差する時間、例えば、図 4 の時間  $t_3$ ) である。トルクコマンドの加速フェーズにおける積分の結果は、 $U_{acc}$  として示される。同様に、適用されたトルクコマンドに応答した動作制御システムから測定された連続した速度データは、加速フェーズにおける速度積分の結果  $V_{acc}$  を得るために時間  $T_0$  から  $T_{CROSSOVER}$  まで積分される。

#### 【0064】

ステップ 1012 では、トルクおよび速度データの減速部分に対して同様の積分を実行する。すなわち、トルクおよび速度データは、動作システムがステップ 1006 において減速し第 2 の速度検出点に戻る時間に対応する時間  $T_{CROSSOVER}$  から  $T_{FINAL}$  20  
まで積分される。ここで、 $T_{FINAL}$  は、減速フェーズの終了時間である。(図 4 に関連した上記の例では、 $T_{FINAL} = t_7$  である。) トルクおよび速度データの減速フェーズにおけるこれらの積分結果は、それぞれ  $U_{dec}$  および  $V_{dec}$  で示される。

#### 【0065】

ステップ 1014 では、動作システムの推定慣性および推定摩擦係数、もしくは、そのいずれかが、積分  $U_{acc}$ 、 $V_{acc}$ 、 $U_{dec}$  および  $V_{dec}$  に基づいて計算される。例えば、推定慣性および推定摩擦係数は、式(12)および(13)もしくはその変形に 30  
基づいて計算されるだろう。

#### 【0066】

##### [動作プロファイル生成器]

図 11 は、動作制御システムの 2 点間移動に対する動作プロファイルを生成することが可能な非限定的な動作プロファイル生成システムを例示するブロック図である。動作プロファイル生成システム 1102 は、ポジションプロファイル生成部 1104、速度プロファイル生成部 1106、インターフェース部 1108、1 以上のプロセッサ 1110 およびメモリ 1112 を含む。いくつかの実施例では、ポジションプロファイル生成部 1104、速度プロファイル生成部 1106、インターフェース部 1108、1 以上のプロセッサおよびメモリ 1112 のうちの 1 つ以上は、動作プロファイル生成システム 1102 の 1 以上の機能を実行するために、電気的および通信可能に、もしくは、そのいずれかであるように相互に結合される。いくつかの実施形態では、ポジションプロファイル生成部 1104、速度プロファイル生成部 1106 およびインターフェース部 1108 は、メモリ 1112 に記憶され、プロセッサ 1110 により実行されるソフトウェア命令を備える。動作 40  
プロファイル生成システム 1102 は、図 11 に表されない他のハードウェアおよびソフトウェアと相互作用しても良い。例えば、プロセッサ 1110 は、キーボード、表示モニタ、タッチスクリーンのようなユーザインターフェース装置と相互作用しても良い。

#### 【0067】

インターフェース 1108 は、ユーザ入力を受け取り、任意の好適な形式(例えば、視覚的、聴覚的、触覚的な形態)でユーザに出力を提供する。ユーザ入力は、例えば、ユーザ設定の制約(例えば、最大加速度、最大速度など)であり、動作プロファイル(以下に、詳細に説明される)を計算するために動作プロファイル生成システム 1102 により使用される。ポジションプロファイル生成部 1104 は、動作システムのために望ましい目標 50  
ポジションの命令を受け取り、ユーザ定義の制約を表すパラメータの範囲内で目標ポジシ

ョンへ移行するための動作プロファイルを計算するように構成される。同じように、速度プロファイル生成部 1106 は、動作制御システムのための望ましい目標速度の命令を受け取り、定められた制約に適合するように現行の速度から目標速度へ動作システムを移行させるための動作プロファイルを生成することができる。図 11 は、ポジションプロファイル生成部 1104 および速度プロファイル生成部 1106 の両方を含む動作プロファイル生成システムを表しているが、動作プロファイル生成システムのいくつかの実施形態は、この開示の範囲から逸脱することなく、ポジションプロファイル生成部 1104 または速度プロファイル生成部のいずれかのみを含むことができることが認識されるべきである。1 以上のプロセッサ 1110 は、開示されたシステムおよび方法、もしくは、そのいずれかを参照して、ここに説明した 1 以上の機能を実行できる。メモリ 1112 は、開示されたシステムおよび方法、もしくは、そのいずれかを参照して、ここに説明した 1 以上の機能を実行するためのコンピュータが実行できる命令および情報、もしくは、そのいずれかを記憶するコンピュータが読み取り可能な記憶媒体である。

10

20

30

40

50

#### 【0068】

いくつかの実施形態では、ここに説明するプロファイル生成部は、動作制御部の集積化された部品であっても良い。図 12 は、この開示の 1 以上の実施形態にしたがってプロファイル生成部 1206 を用いる主制御部 1202 を備えた動作制御システム 1200 を例示している。主制御部 1202 は、例えば、1 以上の動作デバイスを含むシステム（例えば、工業的処理、自動システム、バッチ処理など）を監視し制御するプログラマブルロジックコントローラ（PLC）、または、そのようなコントローラである。この例では、プロファイル生成部 1206 は、主制御部 1202 上に存在する 1 以上のプロセッサにより実行されるコントローラの動作システムおよび制御ソフトウェア、もしくは、そのいずれかの機能部品であっても良い。プロファイル生成部 1206 は、コントローラ 1208 の他の機能要素とデータを交換する回路基板、集積回路など、主制御部 1202 の内部に存在するハードウェア部品であっても良い。プロファイル生成部 1206 の他の好適な実施例も、この開示のいくつかの実施形態の範囲内である。例えば、プロファイル生成部 1206 は、主制御部 1202 に集積化された要素として図 12 に表されているが、いくつかの実施形態では、プロファイル生成部 1206 は、主制御部 1202 から分離された要素であっても良い。そのような構成では、プロファイル生成部 1206 は、主制御部 1202 または動作システムの他の要素と、好適な通信手段を介してデータを交換する。その通信手段は、有線または無線のネットワーク、配線されたデータリンクもしくは、そのような通信手段を含むが、これらに限定される訳ではない。

#### 【0069】

例示的な動作制御システム 1200 は、主制御部 1202 により与えられる動作プロファイル 1212 にしたがって、動作デバイス（例えば、図示しないモータ）を制御するための動作制御部 1214 を含むモータ駆動部 1222 も備える。動作プロファイル 1212 は、現在のポジションまたは速度から目標ポジションまたは目標速度へ動作デバイスの移行させるための軌跡を決定する。ここで、軌跡は、ポジションの基準、速度の基準、加速度の基準およびジャークの基準のうちの 1 つ以上の観点から定められる。主制御部 1202 から動作プロファイルデータを受信することに応じて、モータ制御部 1214 は、動作プロファイル 1212 を制御信号 1216 に変換し、制御信号 1216 は、動作デバイスを目標ポジションまたは目標速度へ移行させるために動作デバイスに送られる。モータ制御部 1214 が閉ループコントローラであれば、制御信号 1216 を出力している間、モータ制御部 1214 は、動作デバイスの現実の状態（例えば、リアルタイムのポジション、速度など）を示すフィードバック信号 1220 も監視する。モータ制御部 1214 は、フィードバック信号 1220 に基づいて、動作デバイスが、動作プロファイル 1212 にしたがって、できるだけ動作プロファイル 1212 に近接して動くように、制御信号 1216 を調整する。また、モータ制御部 1214 が開ループコントローラであれば、モータ制御部 1214 は、動作プロファイル 1212 に基づいて制御信号 1216 を生成し続けるが、その結果として動作デバイスが動作している間、フィードバック信号 1220 を監

視することはない。

#### 【0070】

この例では、主制御部1202は、主制御部1202に記憶され、実行される制御プログラム1210にしたがってシステムを制御する。動作の間、制御プログラム1210は、動作デバイスが新規のポジションへ動くか、または、新規の速度へ移行することを要求する。目的のポジションおよび速度1208は、プロファイル生成部1206に与えられ、プロファイル生成部1206は、動作プロファイル1212を計算し動きの軌跡を決定する。プロファイル生成部1206は、動作システムの機械的制約、または、動作デバイスの動作に関するユーザの好みを表す1以上の動作の制約条件1204の関数として動作プロファイル1212を計算する。動作定数1204は、動作の前に（例えば、図11のインターフェース部1108を介して）ユーザにより与えられる。いくつかの実施形態では、以下に詳細に説明するように、プロファイル生成部1206は、プロファイルのセグメントがコントローラのサンプル点に一致することを確実にするために、付加的に、主制御部1202のサンプリング時間1218に基づいて動作プロファイル1212を計算することもできる。

10

#### 【0071】

以下に、より詳細に説明するように、動作プロファイル1212は、参照ポジションの基準、速度の基準、加速度の基準およびジャークの基準として2点間移動の時間に対する軌跡を決定する。これらの参照値は、動作プロファイル生成部1206により計算された関数を表し、与えられた2点間移動に対する時間の関数として、それぞれの動きの特性がどのように制御されるかを決定する。これらの参照値は、微分係数として数学的な相互関係を有する。すなわち、ジャークは、加速度の微分係数であり、加速度は、速度の微分係数、そして、速度は、ポジションの微分係数である。以下に、詳細に議論するように、プロファイル生成部1206は、軌跡プロファイルのそれぞれの段階に対して、これらの参照値を計算できる。

20

#### 【0072】

一旦、移動のための動作プロファイル1212を計算すれば、プロファイル生成部1206は、動作プロファイル1212をモータ制御部1214に与え、モータ制御部1214は、動作プロファイル1212を制御信号1216に変換する。制御信号1216は、動作プロファイル1212にしたがった望ましい2点間移動の実行を動作デバイスに命令する。前述したように、モータ制御部1214が閉ループコントローラであれば、制御信号1216は、動作プロファイル1212の関数であると同時に、動作デバイスの現実の状態をモータ制御部1214にリアルタイムに知らせるフィードバック信号1220の関数でもある。閉ループコントローラに対しては、制御信号1216は、動作プロファイル1212だけの関数となる。

30

#### 【0073】

図12の表された構成は、プロファイル生成部1206が動作する背景を例示することのみを意図しており、他の動作環境もこの開示に含まれる。例えば、いくつかのシナリオでは、主制御部1202は、モータを制御できる機能を統合したコントローラを包含しても良い。そのような形態では、主制御部1202は、離間したモータ駆動部1222に動作プロファイル1212を与えるよりも、動作プロファイル1212を好適な制御信号316に変換し、制御信号1216を動作デバイスに送る。他の例示的な構造では、プロファイル生成部1206は、モータ駆動部1222に統合されても良い。

40

#### 【0074】

プロファイル生成部1206は、ポジションプロファイル生成部および速度プロファイル生成部のいずれか、もしくは、両方であっても良い。これら2つのタイプのプロファイル生成部は、図13および14にそれぞれ表されている。図13に示すように、ポジションプロファイル生成部1302は、入力として、制御されたシステムの機械的制約、または、動作システムの振る舞いに関するユーザの好みを表す制約のセット1304を受信する。これらの制約は、コントローラの制御信号の更新された周期（通常、ミリ秒単位で測

50

定される)を表すサンプリング時間、速度、加速度、減速度およびジャークの上限を含む。いくつかの実施形態では、ポジションプロファイル生成部1302がコントローラのサンプリング時間を自動的に決定するが、これらの制約は、(例えば、図11のインターフェース部1108を介して)ユーザにより設定しても良い。これらの制約1304は、動作制御システムを準備する間に、一度設定しても良いし、動作ごとに再構成しても良い。ポジションプロファイル生成部1302は、非対称の加速および減速を有するプロファイルをそれぞれ提供する構成とするために加速および減速の限界を許容する。以下に、詳細に説明するように、このサンプリング時間は、プロファイル生成部1302により動作プロファイルの精度を向上させるために用いられる。

【0075】

動作の間、ポジションプロファイル生成部1302は、動作システムに対する新規の目標ポジションを特定するポジションステップコマンド1308を受信する。ポジションステップコマンド1308は、コントローラにより実行される制御プログラム(例えば、図12の制御プログラム)により生成されるか、もしくは、ユーザによりマニュアル入力される動作命令でも良い。ポジションステップコマンド1308に応答して、ポジションプロファイル生成部は、制約に基づく時間的に最適化された動作プロファイル1306を計算する。動作プロファイル1306は、負荷をその現在のポジションからポジションステップコマンド1308により定められた目標ポジションへ移動させるための軌跡を定める。動作プロファイル1306は、ジャーク基準、加速度基準、速度基準およびポジション基準のうちの1以上を含む。これらは、微分係数として数学的な相互関係を有する。ポジションプロファイル生成部1302は、決定される動作プロファイルの段階もしくはセグメントの組み合わせのそれぞれに対し、時間の関数としてこれらの基準を定める。表1は、2点間の動作プロファイルの7つのセグメントをまとめたものである。

【0076】

【表1】

No.	セグメント名	説明
1	ACC_INC	加速度プロファイルはゼロから最大加速度まで増加する
2	ACC_HOLD	加速度プロファイルは最大加速度に保持される
3	ACC_DEC	加速度プロファイルは最大加速度からゼロまで減少する
4	VEL_HOLD	速度プロファイルは一定に保持される
5	DEC_INC	加速度プロファイルの絶対値はゼロから最大減速度まで増加する
6	DEC_HOLD	加速度プロファイルの絶対値は最大減速度に保持される
7	DEC_DEC	加速度プロファイルの絶対値は最大減速度からゼロまで減少する

【0077】

初めに、第1段階(ACC\_INC)の間、加速度は、ゼロから一定の加速度まで連続的に増

加する。いくつかのシナリオでは、この定加速度は、制約条件 1 3 0 4 により定められた最大加速度である。しかしながら、比較的短いポジションステップでは、ポジションプロファイル生成部 1 3 0 2 は、小さな加速度がより正確な目標ポジションへの移行をもたらすことを明らかにするだろう。第 2 段階 (ACC\_HOLD) では、加速度は一定のレートに保持される。システムがポジションプロファイル生成部 1 3 0 2 により計算された目標速度に近づいたとき、一定速度に達するまで加速度が徐々に減少する第 3 段階 (ACC\_DEC) に入る。一定速度に達すると、システムが目標ポジションに近づくように、第 4 段階 (VEL\_HOLD) の間、この一定速度は保持される。システムが目標ポジションに近づくとき、軌跡は第 5 段階に入る。第 5 段階では、システムは、ゼロから動作プロファイルにより定められた目標減速度へ徐々にレートを増加させながら減速を始める。目標の減速度に達すると、第 6 段階 (DEC\_HOLD) の間、この減速度が保持される。最終的に、第 7 段階 (DEC\_DEC) の間、減速度はシステムが速度ゼロに達するまで徐々に減少し、動作シーケンスを終了する。

10

#### 【 0 0 7 8 】

ポジションステップコマンド 1 3 0 8 が与えられると、ポジションプロファイル生成部 1 3 0 2 は、時間的に最適化された動作プロファイルに対し、これら 7 つのプロファイルセグメントのうちのどれが必要かを決定し、時間的に変化するジャーク基準、加速度基準、速度基準およびポジション基準のうちの 1 以上を、動作に必要と思われるそれぞれのセグメントに対して計算する。それぞれの段階に対し計算された基準は、動作プロファイルを完成するために結合される。完結した動作プロファイルは、それにより決定される軌跡を介して動作システムを駆動するために、開ループまたは閉ループの動作コントローラ (例えば、モータ駆動部) で使用することができる。

20

#### 【 0 0 7 9 】

図 1 4 は、1 以上の実施形態にしたがう速度プロファイル生成部 1 4 0 2 の例を表している。速度プロファイル生成部 1 4 0 2 は、ポジションプロファイル生成部 1 3 0 2 に類似するが、ポジションの変化よりも速度の望ましい変化に応答する動作プロファイルを計算するために用いられる。すなわち、速度プロファイル生成部 1 4 0 2 は、現在の速度から速度設定点 1 4 0 8 により特定される目標速度への動作システムの移行に対して、時間的に最適化された動作プロファイル 1 4 0 6 を計算する。望ましい速度設定点への移行は、通常、動作システムのポジションに無関係なので、速度プロファイル生成部 1 4 0 2 に対して定められる制約条件 1 4 0 4 は、ポジションの限界を含まなくても良い。同じように、プロファイル生成部 1 4 0 2 により生成される動作プロファイル 1 4 0 6 は、ポジション基準を含まなくても良く、時間的に変化するジャークの基準、加速度基準および速度基準のうちの少なくとも 1 つの観点から排他的に動作プロファイルを決定しても良い。

30

#### 【 0 0 8 0 】

いくつかの動作制御の用途では、動作コントローラは、台形の動作プロファイル、または、S 曲線のプロファイルのいずれか 1 つを生成する。これらのプロファイルに加えて、または、これらに代えて、この開示に係るプロファイル生成部は、以下、S T プロファイルとして言及する第 3 のタイプのプロファイルを生成できる。図 1 5 は、通常、台形および S 曲線プロファイルと S T 曲線プロファイルの例を比較している。図 1 5 に表された時間グラフは、ポジション 0 (開示ポジション) と、ポジション 2 . 5 (図 1 3 のポジションステップコマンド 1 3 0 8 により決定される目標ポジション) と、の間における与えられた軌跡に対するポジション、速度、加速およびジャークをプロットしたものである。一般的に理解されるように、プロットされた値は、相互に微分係数として数学的に関連する。すなわち、速度は、ポジションの微分 (すなわち、ポジション変化のレート) であり、加速度は速度の微分、そして、ジャークは、加速度の微分である。

40

#### 【 0 0 8 1 】

台形の動作プロファイルは、上記の 7 つの段階のうちの 3 つ、一定加速度 (第 2 段階)、一定速度 (第 4 段階) および一定減速度 (第 6 段階) を用いるだけである。この台形の速度プロファイルの結果は、図 1 5 中の破線の速度曲線により表される。一定加速度段階

50

もしくは一定減速度段階と、一定速度段階と、の間の急峻な以降は、台形の速度曲線の頂部における鋭いコーナをもたす。台形プロファイルの加速および減速フェーズは、常に一定であり、このプロファイルに対する加速度曲線は、加速度グラフに破線で表されるように、一定値の間で階段状に変化する。この例では、減速のレートは加速のレートの2倍であり、台形のケースにおける加速度曲線は、加速段階の間、0.5であり、一定速度段階では、0（ゼロ）であり、減速段階では、-1.0である。また、ジャーク曲線（加速および減速の変化のレートを表す）は、移行の間の短時間のパルス（プロットしない）であり、図15のジャークグラフに破線で表すように加速度または減速度が一定の時、ゼロのままである。

#### 【0082】

台形プロファイルは、段階的な移行なしに、常に、一定速度段階へ一定のレートで加速し、一定速度段階から一定のレートで減速するため、台形曲線プロファイルは、現在のポジションと目標ポジションとの間を比較的早く移動する。しかしながら、加速または減速段階と一定速度段階と、の間の急峻な移行は、システムの望ましくない機械的な乱れを生じさせる。さらに、動作システムが目標ポジションに近づくにつれて徐々に減速することがなく、目標ポジションに到達するまで一定の減速度を維持し、急に速度ゼロに移行するので、台形の動作プロファイルは、最初の移行過程の終わりに目標ポジションを通り過ぎる可能性が高く、コントローラは、負荷を目標ポジションに戻す補償制御信号を出力する必要がある。この過程は、システムが目標ポジションに落ち着くまで数回繰り返され、システムの望ましくない振動を生じさせる。

#### 【0083】

台形プロファイルとは対照的に、S曲線プロファイル（図15のグラフに細い実線で表される）では、7つのプロファイル段階の連続的な加速および減速段階の全てを用いる。これにより、一定（またはゼロ）速度フェーズと、一定加速および減速段階と、の間に段階的な移行が許容される。これらの段階的な加速度の移行は、加速度グラフに明確に表されている。台形プロファイルのように、時間0（ゼロ）から一定の加速度で開始されるのではなく、S曲線プロファイルの加速度は、時間0（ゼロ）から徐々に一定の加速度まで傾斜して増加する。速度が最大（1sと2sとの間）に達した時、加速度は、台形のケースのように急峻にゼロへ減少するのではなく、一定の速度を得るために徐々にゼロまで減少する。加速度の同じような段階的な変化は、S曲線プロファイルの後の減速段階でも見られる。これらの段階的な加速度変化の効果は、速度およびポジションのグラフにそれぞれみられる。具体的には、S曲線の速度プロファイルのコーナが台形カーブに比べて丸められ、加速および減速段階と、一定速度段階と、の間のスムーズな移行を表している。同じように、S曲線のポジションプロファイルは、最初のポジションと、目標ポジションと、の間のスムーズな移行を示しているが、目標に達するための付加的な時間を費やす。

#### 【0084】

ここに説明されるプロファイル生成部の1以上の実施形態は、S曲線の動作プロファイルの生成をサポートする。通常、S曲線の動作プロファイルを用いる動作制御システムは、対照的な加速および減速をサポートするだけである。すなわち、一定の加速度および一定の減速度の絶対値は等しい。これに対し、ここに説明するプロファイル生成部は、非対称の加速度および減速度を有するS曲線の動作プロファイルをサポートすることができる。これは、図15の加速度グラフに表され、加速度に対する0.5の限界と、減速度に対する-1の限界と、を有するS曲線を表している。そのような非対称な加速度と減速度とを与えるために、プロファイル生成部は、システムの制約条件として、異なる加速度および減速度の限界（例えば、図13の制約条件1304を参照）を許容し、これらの制約条件を考慮して動作プロファイルを計算する。

#### 【0085】

ジャークのグラフに表されるように、S曲線のケースにおいて、動作プロファイルの第1、第3、第5および第7段階の間、段階加速度が増加し減速度が減少するレートは、常に同じである。すなわち、この例では、動作プロファイルの任意の段階において、ジャーク

10

20

30

40

50



クは、1、0もしくは-1のいずれかであり、常に一定値である。これは、加速度グラフに表すように、増加または減少する加速（または減速）段階と、一定の加速段階と、の間における急峻な移行をもたらす。

#### 【0086】

ポジション間の時間的に最適化された移行を得る通常の動作制御システムの台形およびS曲線プロファイルよりもスムーズな動作を得るために、ここに説明する1以上の実施形態に係るプロファイル生成部は、ST曲線型のプロファイルにしたがった動作プロファイルを計算することができる。ST曲線プロファイルの例は、図15のグラフに太い実線で表されている。台形およびS曲線プロファイルとは対照的に、ST曲線プロファイルでは、加速度および減速度を増加および減少させる段階において、時間に対するジャークの変化が連続的であり段階的である。これにより、図15の加速度グラフにおける加速度の遷移がスムーズになり、これに対応して速度およびポジションのグラフそれぞれ示す速度およびポジション曲線もスムーズになる。

10

#### 【0087】

さらに、ST曲線プロファイルは、非対称の加速度および減速度をサポートすることができる。（すなわち、プロファイル生成部は、与えられた動作プロファイルに対して減速のレートと異なる加速レートを有するプロファイルを計算することができる。）時間的に最適化された解を見つけると同時に、時間の関数として数学的な軌跡の表現を導出することは、非対称の加速度および減速度を用いる場合の課題である。これを解決するために、ここに説明されるプロファイル生成部の1以上の実施形態は、加速と減速との関係、および、加速時のジャークと、減速時のジャークと、の関係を活用したアルゴリズムを用い、これらの関係を導出の過程に適用する。これにより、軌跡の解析的な表現の導出が可能となり、時間的に最適化された解を見つけることができる。

20

#### 【0088】

ST曲線のポジションプロファイルの例を以下に導出する。ここに説明するプロファイル生成部の1以上の実施形態は、以下の導出過程に基づいて、動作プロファイルの基準を生成することができる。しかしながら、ここに説明するプロファイル生成部は、ST曲線に基づいた動作プロファイルの生成のためのこの技術に限定される訳ではなく、時間の関数として定義される連続したジャーク曲線を得る任意の好適なアルゴリズムは、この開示の範囲に含まれることを理解すべきである。

30

#### 【0089】

次の方程式において、 $\ddot{x}$ 、 $\dot{x}$ 、 $x$  および  $t$  は、それぞれ、ジャーク、加速度、速度およびポジションであり、 $t_1$ 、 $t_2$ 、 $t_3$ 、 $t_4$  および  $t_5$  は、それぞれ、動作プロファイルのACC\_\_INC、ACC\_\_HOLD、VEL\_\_HOLD、DEC\_\_INCおよびDEC\_\_HOLD段階（表1参照）の継続時間である。この例では、ACC\_\_INCおよびACC\_\_DECの継続時間は等しいと仮定される。すなわち、 $t_1$  は、ACC\_\_INCおよびACC\_\_DEC段階の両方の継続時間である。同様に、DEC\_\_INCの継続時間は、DEC\_\_DECと同じであり、 $t_4$  は、DEC\_\_INCおよびDEC\_\_DECの両方の継続時間である。Kは、次の式にしたがって、ジャーク、加速度、速度およびポジションのそれぞれの動作プロファイルにおけるそれぞれの段階に対して定められる利得値である。（ここで、7つのそれぞれの段階に対して、 $t = 0$  は、それぞれの段階の開始時間を表す。）

40

【数 1 4】

$$\ddot{\theta}(t) = K \begin{cases} t(t_1 - t) & 0 \leq t \leq t_1 \\ 0 & 0 \leq t \leq t_2 \\ -t(t_1 - t) & 0 \leq t \leq t_1 \\ 0 & 0 \leq t \leq t_3 \\ -t(t_4 - t) & 0 \leq t \leq t_4 \\ 0 & 0 \leq t \leq t_5 \\ t(t_4 - t) & 0 \leq t \leq t_4 \end{cases} \quad (14)$$

10

【数 1 5】

$$\ddot{\theta}(t) = K \begin{cases} t^2 \frac{3t_1 - 2t}{6} & 0 \leq t \leq t_1 \\ \frac{t_1^3}{6} & 0 \leq t \leq t_2 \\ \frac{t_1^3}{6} + t^2 \frac{2t - 3t_1}{6} & 0 \leq t \leq t_1 \\ 0 & 0 \leq t \leq t_3 \\ t^2 \frac{2t - 3t_4}{6} & 0 \leq t \leq t_4 \\ -\frac{t_4^3}{6} & 0 \leq t \leq t_5 \\ -\frac{t_4^3}{6} + t^2 \frac{3t_4 - 2t}{6} & 0 \leq t \leq t_4 \end{cases} \quad (15)$$

20

30

【数 1 6】

$$\dot{\theta}(t) = K \begin{cases} \frac{t^3}{12}(2t_1 - t) & 0 \leq t \leq t_1 \\ \frac{t_1^3(2t + t_1)}{12} & 0 \leq t \leq t_2 \\ \frac{t_1^3(t_1 + t_2)}{6} - \frac{1}{12}(t + t_1)(t_1 - t)^3 & 0 \leq t \leq t_1 \\ \frac{t_1^3(t_1 + t_2)}{6} & 0 \leq t \leq t_3 \\ \frac{t_1^3(t_1 + t_2)}{6} - \frac{t^3}{12}(2t_4 - t) & 0 \leq t \leq t_4 \\ \frac{t_1^3(t_1 + t_2)}{6} - \frac{t_4^3}{12}(2t + t_4) & 0 \leq t \leq t_5 \\ \frac{t_1^3(t_1 + t_2)}{6} - \frac{t_4^3(t_4 + t_5)}{6} + \frac{(t_4 - t)^3(t + t_4)}{12} & 0 \leq t \leq t_4 \end{cases} \quad (16)$$

【数 1 7】

$$\theta(t) = K \begin{cases} \frac{t^4}{120}(5t_1 - 2t) & 0 \leq t \leq t_1 \\ \frac{t_1^5}{40} + \frac{t_1^3(t + t_1)}{12}t & 0 \leq t \leq t_2 \\ \frac{t_1^3}{12}(t_1 + t_2)(2t + t_2) + \frac{2t + 3t_1}{120}(t_1 - t)^4 & 0 \leq t \leq t_1 \\ \frac{t_1^3(t_1 + t_2)}{12}(2t + 2t_1 + t_2) & 0 \leq t \leq t_3 \\ \frac{t_1^3(t_1 + t_2)}{12}(2t + 2t_3 + 2t_1 + t_2) + \frac{t^5}{60} - \frac{t_4 t^4}{24} & 0 \leq t \leq t_4 \\ \frac{t_1^3(t_1 + t_2)}{12}(2t + 2t_4 + 2t_3 + 2t_1 + t_2) - \frac{t_4^5}{40} - \frac{t_4^3 t^2}{12} - \frac{t_4^4 t}{12} & 0 \leq t \leq t_5 \\ \frac{t_1^3(t_1 + t_2)}{12}(t_5 + 2t_4 + 2t_3 + 2t_1 + t_2) - \frac{(t_4 - t)^4(2t + 3t_4)}{120} & 0 \leq t \leq t_4 \end{cases} \quad (17)$$

【0 0 9 0】

これらの関係が与えられれば、最大加速ジャーク、最大減速ジャーク、最大加速度、最大減速度および最大速度をセグメントの継続時間により表すことができる。

【数 1 8】

$$K=P \frac{12}{t_1^3(t_1+t_2)(t_5+2t_4+2t_3+2t_1+t_2)} \quad (18)$$

【数 1 9】

$$J_e=P \frac{3}{t_1(t_1+t_2)(t_5+2t_4+2t_3+2t_1+t_2)} \quad (19)$$

【数 2 0】

$$I=P \frac{3}{t_4(t_4+t_5)(t_5+2t_4+2t_3+2t_1+t_2)} \quad (20)$$

10

【数 2 1】

$$A=P \frac{2}{(t_1+t_2)(t_5+2t_4+2t_3+2t_1+t_2)} \quad (21)$$

【数 2 2】

$$D=P \frac{2}{(t_4+t_5)(t_5+2t_4+2t_3+2t_1+t_2)} \quad (22)$$

20

【数 2 3】

$$V=P \frac{2}{(t_5+2t_4+2t_3+2t_1+t_2)} \quad (23)$$

ここで、P はポジションステップ、J e は最大加速ジャーク、I は最大減速ジャーク、A は最大加速度、D は最大減速度および V は最大速度である。

【0 0 9 1】

P、V、A、D、J e、I、t<sub>1</sub>、t<sub>2</sub>、t<sub>3</sub>、t<sub>4</sub> および t<sub>5</sub> の間の関係は、次のように得られる。

30

【数 2 4】

$$t_1^3(t_1+t_2) = t_4^3(t_4+t_5) \quad (24)$$

【数 2 5】

$$t_1 = \frac{3A}{2J_e} \quad (25)$$

【数 2 6】

$$t_4 = \frac{3D}{2I} \quad (26)$$

40

【数 2 7】

$$t_2 = \frac{V}{A} - \frac{3A}{2J_e} \quad (27)$$

【数 2 8】

$$t_5 = \frac{V}{D} - \frac{3D}{2I} \quad (28)$$

50

【数 2 9】

$$t_3 = \frac{P}{V} - \frac{3A}{4Je} - \frac{V}{2A} - \frac{3D}{4I} - \frac{V}{2D} \quad (29)$$

【数 3 0】

$$\frac{A^2}{D^2} = \frac{Je^3}{I^3} \quad (30)$$

【0 0 9 2】

10

$t_1$ 、 $t_2$ 、 $t_3$ 、 $t_4$  および  $t_5$  は、全て 0（ゼロ）以上であるとして、次の関係を仮定すれば、

【数 3 1】

$$\frac{D}{A} = \rho \quad (31)$$

【数 3 2】

$$\frac{I}{Je} = \rho^{\frac{2}{3}} \quad (32)$$

20

次の不等式が成立する。

【数 3 3】

$$V \geq \frac{3A^2}{2Je} = \frac{3}{2I\rho^{\frac{4}{3}}} D^2 \quad (33)$$

【数 3 4】

$$V \geq \frac{3}{2I} D^2 \quad (34)$$

30

【数 3 5】

$$\frac{P}{V} \geq \frac{3}{4I} \rho^{-\frac{1}{3}} D + \frac{\rho}{2} \frac{V}{D} + \frac{3}{4I} D + \frac{1}{2} \frac{V}{D} \quad (35)$$

【数 3 6】

$$\frac{P}{V} \geq \left( \rho^{-\frac{1}{3}} + 1 \right) \frac{3}{4I} D + \frac{1}{2} \left( \rho + 1 \right) \frac{V}{D} \quad (36)$$

不等式（33）～（36）を解くことにより、 $V$ 、 $A$ 、 $D$ 、 $Je$  および  $I$ （速度、加速  
度、加速ジャークおよび減速ジャークのそれぞれの最大値）の適切な値を得る。

40

【0 0 9 3】

これらの最大値を式（25）～（29）に代入することにより、 $t_1$ 、 $t_2$ 、 $t_3$ 、 $t_4$  および  $t_5$  の値を得ることができる。上記の式（14）～（36）により導出される  $V$ 、 $A$ 、 $D$ 、 $Je$ 、 $I$ 、 $t_1$ 、 $t_2$ 、 $t_3$ 、 $t_4$  および  $t_5$  の値は、定められた機械的な制約条件もしくはユーザの要求の範囲内で動作する時間的に最適化されたスムーズな軌跡を与える。

【0 0 9 4】

上記の関係に基づいて、プロファイル生成部は、与えられた 2 点間の移動に対して、好適な ST 曲線の動作プロファイルを計算することができる。しかしながら、 $t_1$ 、 $t_2$ 、

50

$t_3$ 、 $t_4$  および  $t_5$  に対して最初に計算される値は、コントローラのサンプリング時間の倍数ではなく、結果として動作コントローラのサンプル点に一致しないであろう。プロファイルセグメントの継続時間がコントローラの2つのサンプル時間の間に収まる場合は、コントローラは、制御信号の望ましい出力と、制御信号の実際の出力と、の間の小さな違いを補正する必要があるだろう。この点を解決するために、ここに説明されるプロファイル生成部の1以上の実施形態は、上記のように、最大値  $V$ 、 $A$ 、 $D$ 、 $J_e$  および  $I$  と、セグメントの継続時間  $t_1$ 、 $t_2$ 、 $t_3$ 、 $t_4$  および  $t_5$  と、を導出した後に、付加的な計算を実施する。

#### 【0095】

具体的には、プロファイル生成部が上記の手順にしたがって  $t_1$ 、 $t_2$ 、 $t_3$ 、 $t_4$  および  $t_5$  を計算した後、それぞれの継続時間を最も近いサンプリング時間に上方修正し、 $t_{1\text{New}}$ 、 $t_{2\text{New}}$ 、 $t_{3\text{New}}$ 、 $t_{4\text{New}}$  および  $t_{5\text{New}}$  を得る。この修正ステップは、制約条件 1304 または 1404 の1つとしてプロファイル生成部に与えられたサンプリング時間に基づくことができる。そして、プロファイル生成部は、修正された継続時間  $t_{1\text{New}}$ 、 $t_{2\text{New}}$ 、 $t_{3\text{New}}$ 、 $t_{4\text{New}}$  および  $t_{5\text{New}}$  の値を用いて、 $V$ 、 $A$ 、 $D$ 、 $J_e$  および  $I$  の新しい値を計算する。この計算により、サンプリング時間の倍数であるセグメントの継続時間を含む最終的な動作プロファイルが得られ、コントローラから出力される制御信号がコントローラのサンプル点に一致することが保証される。これにより、動作プロファイルの時間が2つのサンプル点の間に収まる場合に生じる小さな違いの補償の必要性を軽減できる。

#### 【0096】

上記のST曲線の代わりに、または、それに加えて、ここに説明するプロファイル生成部の1以上の実施形態は、非対称の加速度および減速度を有するS曲線プロファイルを生成することが可能である。非対称の加速度および減速度を有するS曲線プロファイルの例は、以下に導出される。ここに説明するプロファイル生成部の1以上の実施形態は、以下の手順またはその変形に基づいて動作プロファイルの基準を生成することができる。

#### 【0097】

上記のST曲線の方程式と同じように、 $\dot{x}$ 、 $\ddot{x}$ 、 $\ddot{\dot{x}}$  および  $\ddot{\ddot{x}}$  は、それぞれジャーク、加速度、速度およびポジションである。 $t_1$ 、 $t_2$ 、 $t_3$ 、 $t_4$  および  $t_5$  は、動作プロファイルのACC\_\_INC、ACC\_\_HOLD、VEL\_\_HOLD、DEC\_\_HOLD、DEC\_\_INC および DEC\_\_HOLDの各段階（表1参照）におけるそれぞれの継続時間である。例示したST曲線と同じように、ACC\_\_INC および ACC\_\_DECの継続時間は同じであり、 $t_1$  は、ACC\_\_INC および ACC\_\_DECの両方の段階における継続時間である。同じように、DEC\_\_INCの継続時間はDEC\_\_DECと同じであり、 $t_4$  は、DEC\_\_INC および DEC\_\_DECの両方の継続時間である。 $K$  は、次の式にしたがって、ジャーク、加速度、速度およびポジションのそれぞれに対する動作プロファイルのそれぞれの段階に対して決定される利得値である。（ここで、7つの段階のそれぞれに対し、 $t = 0$  は、それぞれの段階の開始時間を表す。）

【数 3 7】

$$\ddot{\theta}(t) = K \begin{cases} 1, & 0 < t < t_1 \\ 0, & 0 < t < t_2 \\ -1, & 0 < t < t_1 \\ 0, & 0 < t < t_3 \\ -1, & 0 < t < t_4 \\ 0, & 0 < t < t_5 \\ 1, & 0 < t < t_4 \end{cases} \quad (37)$$

10

【数 3 8】

$$\ddot{\theta}(t) = K \begin{cases} t, & 0 < t < t_1 \\ t_1, & 0 < t < t_2 \\ t_1 - t, & 0 < t < t_1 \\ 0, & 0 < t < t_3 \\ -t, & 0 < t < t_4 \\ -t_4, & 0 < t < t_5 \\ t - t_4, & 0 < t < t_4 \end{cases} \quad (38)$$

20

【数 3 9】

$$\dot{\theta}(t) = K \begin{cases} \frac{1}{2}t^2, & 0 < t < t_1 \\ t_1 \left( t + \frac{1}{2}t_1 \right), & 0 < t < t_2 \\ \frac{t_1}{2}(2t_2 + t_1) + t_1 t - \frac{1}{2}t^2, & 0 < t < t_1 \\ t_1(t_2 + t_1), & 0 < t < t_3 \\ t_1(t_2 + t_1) - \frac{1}{2}t^2, & 0 < t < t_4 \\ t_1(t_2 + t_1) - \frac{1}{2}t_4^2 - t_4 t, & 0 < t < t_5 \\ t_1(t_2 + t_1) - t_4(t_4 + t_5) + \frac{1}{2}(t_4 - t)^2, & 0 < t < t_4 \end{cases} \quad (39)$$

30

40

【数 4 0】

$$\theta(t) = K \begin{cases} \frac{1}{6}t^3, & 0 < t < t_1 \\ \frac{1}{6}t_1^3 + \frac{1}{2}t_1(t+t_1)t, & 0 < t < t_2 \\ \frac{1}{2}t_1(t_2+t_1)(t_2+2t) + \frac{1}{6}(t_1-t)^3, & 0 < t < t_1 \\ \frac{1}{2}t_1(t_1+t_2)(2t_1+t_2+2t), & 0 < t < t_3 \\ \frac{1}{2}t_1(t_1+t_2)(2t_1+t_2+2t_3+2t) - \frac{1}{6}t^3, & 0 < t < t_4 \\ \frac{1}{2}t_1(t_1+t_2)(2t_1+t_2+2t_3+2t_4+2t) - \frac{1}{6}t_4^3 - \frac{1}{2}t_4^2t - \frac{1}{2}t_4t^2, & 0 < t < t_5 \\ \frac{1}{2}t_1(t_1+t_2)(2t_1+t_2+2t_3+2t_4+t_5) - \frac{1}{6}(t_4-t)^3, & 0 < t < t_4 \end{cases} \quad (40)$$

【0 0 9 8】

これらの関係が与えられれば、最大加速ジャーク、最大減速ジャーク、最大加速度、最大減速度および最大速度は、セグメントの継続時間により表される。

【数 4 1】

$$K = P \frac{2}{t_1(t_1+t_2)(t_5+2t_4+2t_3+2t_1+t_2)} \quad (41)$$

【数 4 2】

$$J_e = I = K \quad (42)$$

【数 4 3】

$$A = P \frac{2}{(t_1+t_2)(t_5+2t_4+2t_3+2t_1+t_2)} \quad (43)$$

【数 4 4】

$$D = P \frac{2}{(t_4+t_4)(t_5+2t_4+2t_3+2t_1+t_2)} \quad (44)$$

【数 4 5】

$$V = P \frac{2}{t_5+2t_4+2t_3+2t_1+t_2} \quad (45)$$

ここで、P はポジションステップ、J<sub>e</sub> は最大加速ジャーク、I は最大減速ジャーク、A は最大加速度、D は最大減速度、および、V は最大速度である。

【0 0 9 9】

P、V、A、D、J<sub>e</sub>、I、t<sub>1</sub>、t<sub>2</sub>、t<sub>3</sub>、t<sub>4</sub> および t<sub>5</sub> の間の関係は、次のように得られる。



【数 4 6】

$$t_1(t_1+t_2) = t_4(t_4+t_5) \quad (46)$$

【数 4 7】

$$t_1 = \frac{A}{Je} \quad (47)$$

【数 4 8】

$$t_2 = \frac{V}{A} - \frac{A}{Je} \quad (48)$$

10

【数 4 9】

$$t_4 = \frac{D}{Je} \quad (49)$$

【数 5 0】

$$t_5 = \frac{V}{D} - \frac{D}{Je} \quad (50)$$

20

【数 5 1】

$$t_3 = \frac{P}{V} - \frac{A}{2Je} - \frac{V}{2A} - \frac{D}{2J} - \frac{V}{2D} \quad (51)$$

$t_1$ 、 $t_2$ 、 $t_3$ 、 $t_4$  および  $t_5$  は、全て 0 (ゼロ) 以上であるとして、次の関係を仮定すれば、

【数 5 2】

$$\frac{D}{A} = \rho \quad (52)$$

30

次の不等式が成立する。

【数 5 3】

$$V \geq \frac{A^2}{Je} = \frac{1}{Je\rho^2} D^2 \quad (53)$$

【数 5 4】

$$V \geq \frac{1}{Je} D^2 \quad (54)$$

40

【数 5 5】

$$\frac{P}{V} \geq \frac{1}{2Je} \rho^{-1} D + \frac{\rho}{2} \frac{V}{D} + \frac{1}{2Je} D + \frac{1}{2} \frac{V}{D} \quad (55)$$

【数 5 6】

$$\frac{P}{V} \geq \left( \rho^{-1} + 1 \right) \frac{1}{2Je} D + \frac{1}{2} \left( \rho + 1 \right) \frac{V}{D} \quad (56)$$

【0 1 0 0】

50

不等式(53)～(56)を解くことにより、S曲線プロファイルに対するV、A、D、J<sub>e</sub>およびI(速度、加速度、現速度、加速ジャークおよび減速ジャークのそれぞれの最大値)の適切な値が得られる。

#### 【0101】

これらの最大値を式(47)～(51)に代入することにより、 $t_1$ 、 $t_2$ 、 $t_3$ 、 $t_4$ および $t_5$ (S曲線動作プロファイルのそれぞれのセグメントの継続時間)の値を得ることができる。上記の式(37)～(56)にしたがって導出されたV、A、D、J<sub>e</sub>、I、 $t_1$ 、 $t_2$ 、 $t_3$ 、 $t_4$ および $t_5$ の値により、非対称の加速度および減速度を有し、定められた機械的制約条件もしくはユーザの要求する範囲内で動作するS曲線プロファイルを創出できる。

10

#### 【0102】

いくつかの実施形態では、プロファイル生成部は、ST曲線プロファイルに関連して前述した付加的な計算に類似した方法により、結果的に得られたS曲線動作プロファイルをコントローラのサンプリング時間に適合させる。すなわち、上記の手順にしたがって $t_1$ 、 $t_2$ 、 $t_3$ 、 $t_4$ および $t_5$ を計算した後、プロファイル生成部は、それぞれの継続時間を最も近いサンプリング時間に合わせて伸ばし、 $t_{1\text{New}}$ 、 $t_{2\text{New}}$ 、 $t_{3\text{New}}$ 、 $t_{4\text{New}}$ および $t_{5\text{New}}$ を得る。そして、プロファイル生成部は、修正された継続時間 $t_{1\text{New}}$ 、 $t_{2\text{New}}$ 、 $t_{3\text{New}}$ 、 $t_{4\text{New}}$ および $t_{5\text{New}}$ の値を用いて、V、A、D、J<sub>e</sub>およびIの新しい値を計算する。

20

#### 【0103】

動作プロファイルは、通常、表1にリストされた7つの段階を含むが、いくつかの2点間の移動では、7つの全てのセグメントを要求しなくても良い。例えば、動作システムの現在の状態と、目標の状態と、の間の距離が比較的短いとき、その動作システムのVEL—HOLD(一定速度)セグメントを省略しても良い。したがって、ここに説明するプロファイル生成部の1以上の実施形態では、自動的あるいは機能的にセグメントのスキップを行っても良い。すなわち、プロファイルの7つの段階の全てについて計算を行うのではなく、最終的な軌跡において、1以上の段階が用いられないのであれば、ここに説明するプロファイル生成部のいくつかの実施形態では、与えられた2点間の移動に対して、最終的な軌跡に用いられる段階についてのみ計算を行うことができる。

30

#### 【0104】

動作プロファイルは、新規の動作コマンドを受けた時、動作プロファイルの計算の間どのセグメントをスキップできるかを自動的に決定する。いくつかの実施形態では、プロファイル生成部は、現在のポジションと目標ポジションとの間(ポジション変更の場合)もしくは、現在の速度と目標速度との間(速度変更の場合)の全体の差にある程度基づいて、どのセグメントをスキップして良いかを定めることができ、現在の状態と目標状態との間の差が小さいほど、動作プロファイルのセグメントのいくつかを省略することが可能となる。そのような実施形態では、現在の状態と目標状態との間の差は、複数の定められた差のレンジと比較され、それぞれの定められた差のレンジは、対応する動作プロファイルから省かれた1以上のセグメントに関連する。

40

#### 【0105】

ここに説明されるプロファイル生成部の1以上の実施形態は、動作データの履歴に基づいて、どのセグメントをスキップできるかを推定しても良い。例えば、プロファイル生成部は、出力された動作コマンドの履歴、および、そのコマンドに回答した動作に対応する軌跡のデータ(例えば、ポジション、速度、加速度およびジャーク、もしくは、そのいずれかの時間に対するデータ)を記録しても良い。プロファイル生成部は、動作の特定のタイプに対し、どのセグメントが省略可能であるかに関して推測を行うためにこの履歴データを解析することができる。したがって、新規の2点間の移動コマンドを受け取った時、その移動に対する動作プロファイルを計算する前に、プロファイル生成部は、類似した特性(例えば、横断すべき距離の類似、動作コマンドを受け取った時の速度の類似など)を有する過去の動作コマンドに回答して実行された軌跡の形状に基づいて、どのセグメント

50

をスキップできるかを推定することができる。

【0106】

図16～図19は、例示した7つの段階のポジションプロファイルのセグメントスキップを表している。図16は、7つの全ての段階を用いるS曲線プロファイルの例を表している。図17は、セグメント4（一定速度段階）をスキップしたプロファイルを表している。プロファイル生成部は、横断されるポジションまたは速度ステップが十分小さく、目標ポジションまたは目標速度に達する前に一定速度段階に達しない場合にそのようなプロファイルを計算できる。そのようなポジションステップまたは速度設定点のコマンドを受けた場合、プロファイル生成部は、望ましい動作のためのプロファイル計算を実施する前に、この決定をすることができ、1～3段階および5～7段階についての計算だけを実行する。同じように、図18は、セグメント2および6をスキップしたプロファイルの例を表し、図19は、セグメント2、4および6をスキップしたプロファイルの例を表している。

10

【0107】

図20～図22は、いくつかの開示された態様にしたがった手順を表している。説明の簡略化のために、手順は一連の行為として示され、説明されるが、開示された態様は、その行為の順に限定されず、いくつかの行為は、ここに示され、説明されたものと異なる順で、他の行為と同時にも行われても良いし、そのいずれかであっても良いことを理解し、認識すべきである。例えば、当業者であれば、手順は、状態図のように相互に関連する状態もしくは事象の連続として選択可能に表されることを理解し、認識するであろう。さらに、いくつかの開示された態様にしたがって手順を実行するために、表示された全ての行為が求められる訳ではない。なお、この開示を通じて以下に示される手順は、そのような手順のコンピュータへの転送を容易にする製造物に記憶させることが可能であることを、さらに認識すべきである。

20

【0108】

図20は、動作制御システムにおける2点間の移動に対する動作プロファイルを計算するための手順例2000を表している。ステップ2002では、動作制御システムに対し、動作の制約条件が定められる。これらの制約条件は、動作制御システムにより制御される機械システムの物理的な制約を表し、速度、加速度、減速度およびジャークの限界を含むことができる。制約条件は、機械システムを制御するために用いられる動作コントローラのサンプリング時間の定義を含むことができる。ステップ2002では、制御される機械システムを新規のポジションまたは速度に移行させるコマンドを受信する。このコマンドは、動作コントローラにより実行される動作制御プログラムに起因しても良いし、ユーザにより入力されるマニュアルの動作コマンドであっても良い。そのコマンドは、動作コントローラに関連するプロファイル生成部（例えば、上記のプロファイル生成部1206、1302もしくは1402）が受信しても良い。

30

【0109】

ステップ2006では、ステップ2004で受信されたコマンドの受け取りに応答して、現在のポジションまたは速度から、コマンドにより指示された新規のポジションまたは速度へ機械システムを動かすために、動作プロファイルが計算される。プロファイル生成部は、動作プロファイルのセグメントの少なくとも1つに対し、時間の関数として定められる連続的なジャーク基準を含むように、この動作プロファイルを計算できる。いくつかの実施形態では、動作プロファイルは、ST曲線として、式(14)～(36)に関連した上記の手順にしたがって計算できる。そのような動作プロファイルは、図15に太い実線で表されたジャークグラフにより描かれた一般的な形式を有するジャーク基準を与え、ジャークは、計算されたジャーク関数にしたがって、最大値と最小値の間で時間に対して緩やかに変化する。ステップ2008では、制御される機械システムは、ステップ2006で定められる動作プロファイルにしたがって、現在のポジションまたは速度から新規のポジションまたは速度に移行するよう命令される。これに伴って、例えば、ステップ2006で計算された動作プロファイルがモータ駆動部に与えられ、その動作プロファイル、

40

50

および、機械システムの状態を表すリアルタイムのデータを与えるフィードバック信号にしたがって機械システムを駆動するモータを制御する。

【0110】

図21は、動作コントローラのサンプリング時間に同調し、制約条件に基づいた時間的に最適化された動作プロファイルを計算するための手順例2100を表している。ステップ2102では、制御される機械システムに対する制約条件が定められる。前述した例のように、これらは、コントローラのサンプリング時間と、速度、加速度、減速度およびジャークの限界と、を含む。これらの制約条件は、コントローラに関連するプロファイル生成部（例えば、上記のプロファイル生成部1206、1302または1402）に与えることができる。ステップ2104では、機械システムを現在のポジションまたは速度から新規のポジションまたは速度に移行させるコマンドを（例えば、プロファイル生成部が）受信する。ステップ2106では、最大加速ジャーク（ $J_e$ ）、最大減速ジャーク（ $I$ ）、最大加速度（ $A$ ）、最大減速度（ $D$ ）および最大速度（ $V$ ）の少なくとも1つを動作プロファイルのそれぞれのセグメントに対する時間の関数として計算することにより、そのコマンドに応答して機械システムの軌跡を制御するための動作プロファイルが生成される。（ここで、プロファイルは、表1に定義された7つのセグメントを含むことができる。）さらに、プロファイルのそれぞれのセグメントの継続時間が（例えば、式（14）～（36）に関連した上記の技術を用いて）計算される。

【0111】

ステップ2108では、ステップ2106で計算された全てのプロファイルセグメントの継続時間がコントローラのサンプリング時間の倍数であるかどうかを判定する。全てのセグメントにおいて継続時間がサンプリング時間の倍数であれば、ステップ2114に移り、プロファイルセグメントの継続時間およびステップ2106で計算された $J_e$ 、 $I$ 、 $A$ 、 $D$ および $V$ の値に基づいて、動作プロファイルを生成する。一方、1以上のプロファイルセグメントがサイクル時間の倍数に等しくなければ、ステップ2110に移り、全てのプロファイルセグメントの継続時間をサンプリング時間の最も近い倍数に上方修正する。ステップ2112では、ステップ2110において修正されたプロファイルセグメントの継続時間に基づいて、 $J_e$ 、 $I$ 、 $A$ 、 $D$ および $V$ のうちの1以上の値が再計算される。ステップ2114では、修正されたプロファイルセグメントの継続時間、および、再計算された $J_e$ 、 $I$ 、 $A$ 、 $D$ および $V$ 、もしくは、そのいずれかに基づいて、動作プロファイルが生成される。

【0112】

図22は、2点間の移動に対する動作プロファイルを、セグメントスキップを用いて効果的に計算する手順例2200を表している。ステップ2202では、制御される機械システムを新規のポジションまたは新規の速度に移行させるコマンドを（例えば、プロファイル生成部1206、1302または1402において）受信する。ステップ2204では、要求された2点間の移動を実施するために、動作プロファイルの7つのセグメントのうちのどれが必要とされるかを決定する。この決定は、例えば、現在のポジションと望ましいポジションとの間の移動すべき距離（ポジション変更の場合）、もしくは、現在の速度と望ましい速度との間の差（速度変更の場合）の測定に基づいて、プロファイル生成部により自動的に行われる。

【0113】

ステップ2206では、ステップ2204の決定に基づいて、望ましい動作を実行するために、7つの全てのセグメントが必要かどうかを判断する。全てのセグメントが求められる場合には、ステップ2210に移動し、7つの全てのセグメントに対してプロファイルの計算を実施することにより、2点間の移動のための動作プロファイルを生成する。一方、ステップ2206において、1以上のプロファイルセグメントが要求されないと判断されれば、ステップ2208に移動し、ステップ2204で決定されたように、必要なセグメントのみについて計算を実施することにより、2点間の移動のための動作プロファイルを生成する。手順例2200にしたがったセグメントスキップは、不必要なプロファイ

ルセグメントの計算に関連した余分な処理を減らすことにより、制約条件に基づいた時間的に最適化された動作プロファイルの計算をより効率的に促進する。

【0114】

[ネットワーク化および分散化された環境の例]

当業者であれば、ここに説明されたいくつかの実施形態は、コンピュータネットワークの一部、または、分散化された演算環境に配置され、媒体を含む任意のデータ記憶装置に接続された任意のコンピュータ、もしくは、クライアント装置、サーバ装置を用いて実施できることを認識するだろう。これに関し、ここに説明する画像編集システムのいくつかの実施形態は、任意の数のメモリまたは記憶装置（例えば、図2のメモリ216もしくは図11のメモリ1112）、および、任意の数の記憶装置を用いる任意の数のアプリケーションおよびプロセスを有する任意のコンピュータシステムまたはコンピュータ環境で実施できる。これは、ネットワーク環境もしくは分散された演算環境に配置された、リモートもしくはローカルの記憶装置を有するサーバコンピュータおよびクライアントコンピュータを備える環境を含むが、それに限定される訳ではない。例えば、図2を参照すれば、トルクコマンド生成部204、速度監視部206、慣性演算部208、摩擦係数演算部210およびインターフェース部212は、1つのデバイスに関連する1つのメモリ216に記憶することができる。また、多数のデバイスのそれぞれに関連した多数のメモリに分散しても良い。同じように、トルクコマンド生成部204、速度監視部206、慣性演算部208、摩擦係数演算部210およびインターフェース部212は、1つのプロセッサ214で実行しても良いし、多数のデバイスに関連した多数のプロセッサを用いて実行しても良い。

10

20

【0115】

計算の分散化は、演算装置および演算システム間の通信によりコンピュータリソースおよびコンピュータサービスを共有する。これらのリソースおよびサービスは、目的とする情報、キャッシュおよびディスクの記憶の交換を含む。これらのリソースおよびサービスは、負荷のバランス、リソースの拡張、処理の専用化などに対する多数のプロセスユニットの処理能力の共有も含む。計算の分散化は、ネットワークの接続性に利点を有し、企業全体を利するためにクライアントが総体の力を利用することを可能とする。これに関し、いくつかのデバイスは、この開示のいくつかの実施形態に寄与することが可能なアプリケーション、オブジェクト、または、リソースを有する。

30

【0116】

図23は、ネットワーク化および分散化された演算環境を例示する模式図である。分散化された演算環境は、演算オブジェクト2310、2312など、および、演算オブジェクトもしくは演算装置2320、2322、2324、2326、2328など、を含み、それぞれは、アプリケーション2330、2332、2334、2336、2338により表されるように、プログラム、方法、データ記憶装置、プログラマブルロジックなどを含む。演算オブジェクト2310、2312など、および、演算オブジェクトもしくは演算装置2320、2322、2324、2326、2328などは、PDA (personal digital assistant)、聴覚および視覚デバイス、携帯電話、MP3プレイヤー、パソコン、ノート型パソコン、タブレットなどを含み、ここに説明する慣性推定器の実施形態は、そのようなデバイスの上に存在し、また、そのようなデバイスと相互に作用する。

40

【0117】

演算オブジェクト2310および2312など、および、演算オブジェクトもしくは演算装置2320、2322、2324、2326、2328などのそれぞれは、演算オブジェクト2310および2312など、および、演算オブジェクトもしくは演算装置2320、2322、2324、2326、2328などのうちの他の1以上のデバイスと、通信ネットワーク2340を経由して、直接または間接的に通信することができる。図23には、1つの要素として表されているが、通信ネットワーク2340は、図23のシステムにサービスを提供する他の演算オブジェクトおよび演算装置を含んでも良いし、多重に相互接続された図示しないネットワークであっても良い。演算オブジェクト2310お

50

よび 2 3 1 2 など、および、演算オブジェクトもしくは演算装置 2 3 2 0、2 3 2 2、2 3 2 4、2 3 2 6、2 3 2 8 などのそれぞれは、この開示のいくつかの実施形態との通信、もしくは、その実行に適した API もしくは、他の対象物、ソフトウェア、ファームウェアおよびハードウェア、もしくは、そのいずれかを使用する、アプリケーション 2 3 3 0、2 3 3 2、2 3 3 4、2 3 3 6、2 3 3 8（例えば、慣性推定器 2 0 2、動作プロファイル生成システム 1 1 0 2 もしくはその要素など）のようなアプリケーションプログラムを含むこともできる。

#### 【0 1 1 8】

分散化された演算環境をサポートするいくつかのシステム、要素およびネットワークの構成がある。例えば、演算システムは、有線もしくは無線システム、局在するネットワークもしくは広く分散されたネットワークにより一体に結合できる。現在、多くのネットワークは、演算を広く分散化できる社会基盤として提供され、多くの異なるネットワークを包含するインターネットに接続される。そして、いくつかの実施形態において、ここで説明されるように、システムに付随する例示的な通信に、任意の好適なネットワーク基盤を用いることができる。

10

#### 【0 1 1 9】

このように、ネットワークトポロジーおよびネットワーク基盤のホストとして、クライアント/サーバ、P 2 P (peer-to-peer) またはハイブリッド構造が用いられる。クライアントは、他の階層もしくはグループのサービスを利用する階層もしくはグループのメンバーである。クライアントは、コンピュータの処理、例えば、大まかには、他のプログラムまたは処理により与えられるサービスを求める命令もしくはタスクのセットであっても良い。クライアントの処理は、他のプログラムもしくはサービスそのものについての全てのワークの詳細を知ることなく、要求したサービスを使用する。

20

#### 【0 1 2 0】

クライアント/サーバ構造、特に、ネットワークシステムでは、クライアントは、他のコンピュータ、例えば、サーバにより提供され、共有されるネットワークリソースにアクセスするコンピュータであっても良い。任意のコンピュータは、環境に依存して、クライアント、サーバもしくはその両方と考えることができるが、図 2 3 では、非限定的な例として、演算オブジェクトもしくは演算装置 2 3 2 0、2 3 2 2、2 3 2 4、2 3 2 6、2 3 2 8 などは、クライアントと考えることができ、演算オブジェクト 2 3 1 0、2 3 1 2 などは、サーバと考えることも良い。ここで、演算オブジェクト 2 3 1 0、2 3 1 2 は、クライアント演算オブジェクトもしくはクライアント演算装置 2 3 2 0、2 3 2 2、2 3 2 4、2 3 2 6、2 3 2 8 などからデータを受信し、データを保存し、データを処理し、クライアント演算オブジェクトもしくはクライアント演算装置 2 3 2 0、2 3 2 2、2 3 2 4、2 3 2 6、2 3 2 8 などにデータを送信するようなデータサービスを提供する。1 以上の実施形態に対してここで説明するように、これらの演算装置のいずれもが、データを処理し、システムの技術に関連するトランザクションサービスまたはタスクを要求しても良い。

30

#### 【0 1 2 1】

サービスは、通常、インターネットまたは無線ネットワークインフラのような、遠隔もしくはローカルネットワークを介してアクセスできるリモートコンピュータシステムである。クライアント処理は、第 1 のコンピュータシステムにおいて利用可能であり、サーバ処理は、第 2 のコンピュータシステムにおいて利用可能である。第 1 のコンピュータシステムおよび第 2 のコンピュータシステムは、通信媒体を介して相互に通信する。したがって、分散化された機能を提供し、多くのクライアントにサーバの情報収集の能力を利用させることができる。ここに説明する技術に準拠して用いられる任意のソフトウェアは、独立して提供しても良いし、多数の演算装置もしくは演算オブジェクトに分散されても良い。

40

#### 【0 1 2 2】

例えば、通信ネットワーク/バス 2 3 4 0 がインターネットであるネットワーク環境で

50

は、演算オブジェクト 2 3 1 0、2 3 1 2 などは、ウェブサーバ、ファイルサーバ、メディアサーバなどであっても良く、クライアント演算オブジェクトもしくは演算装置 2 3 2 0、2 3 2 2、2 3 2 4、2 3 2 6、2 3 2 8 などは、H T T P (hypertext transfer protocol) のような既知のプロトコルを介して通信する。演算オブジェクト 2 3 1 0、2 3 1 2 などは、分散化された演算環境を特徴づけるように、クライアント演算オブジェクトもしくはクライアント演算装置 2 3 2 0、2 3 2 2、2 3 2 4、2 3 2 6、2 3 2 8 のように機能しても良い。

#### 【 0 1 2 3 】

##### [ 演算装置の例示 ]

前述したように、ここに説明する技術は、有利に任意の好適なデバイスに適用できる。このため、いくつかの実施形態に関連して、携帯可能な演算装置および他の演算装置、演算オブジェクトなどのすべての種類の使用が期待されることを理解すべきである。したがって、次の図 2 4 で説明するコンピュータは、その 1 例である。さらに、好適なサーバは、メディアサーバもしくは他のメディア管理サーバの要素のような、以下のコンピュータの 1 以上の態様を含むことができる。

10

#### 【 0 1 2 4 】

要求されることはないが、実施形態は、装置もしくは器機のサービスの開発者に使用されるために、オペレーティングシステムを介して部分的に実行されても良いし、ここに説明されるいくつかの実施形態の 1 以上の機能的な態様を実施するために動作するアプリケーションソフトウェアの範囲に含まれても良い。ソフトウェアは、プログラムモジュールのように、コンピュータが実行可能な命令の一般的な背景において説明しても良く、クライアントワークステーション、サーバまたは他の装置のような、1 以上のコンピュータにより実行される。コンピュータシステムは、様々な構成とデータ通信に用いる様々なプロトコルを有するため、特定の構成およびプロトコルに限定されるべきではないことを、当業者は認識するだろう。

20

#### 【 0 1 2 5 】

図 2 4 は、ここに説明する実施形態の 1 以上の態様を実施可能な、好適な演算システム環境 2 4 0 0 の例を表している。上記のように、演算システム環境 2 4 0 0 は、好適な演算環境の 1 つの例に止まり、使用および機能性の範囲についていかなる限定も意図していない。例示的な演算システム環境 2 4 0 0 に表された要素のいずれか、または、その結合に関連するいかなる依存性も、要請も、演算システム環境 2 4 0 0 は有しないと解釈される。

30

#### 【 0 1 2 6 】

図 2 4 を参照して、コンピュータ 2 4 1 0 の形で 1 以上の実施形態を実施する例示的な演算装置を説明する。コンピュータ 2 4 1 0 の構成要素は、プロセスユニット 2 4 2 0、システムメモリ 2 4 3 0、および、システムメモリを含むいくつかの構成要素をプロセスユニット 2 4 2 0 に結合するシステムバス 2 4 2 2 を含むが、これに限定される訳ではない。プロセスユニット 2 4 2 0 は、例えば、慣性推定器 2 0 2 のプロセッサに関連する機能を実行し、システムメモリ 2 4 3 0 は、メモリ 2 1 6 に関連する機能を実行する。

40

#### 【 0 1 2 7 】

コンピュータ 2 4 1 0 は、通常、コンピュータが読み取り可能ないくつかの媒体、すなわち、コンピュータ 2 4 1 0 がアクセス可能な任意の利用可能な媒体を含む。システムメモリ 2 4 3 0 は、R O M (read only memory) および R A M (random access memory) のような揮発性および不揮発性メモリ、もしくは、そのいずれかの形のコンピュータ記憶媒体を含むことができる。限定ではなく例示の形で、システムメモリ 2 4 3 0 は、オペレーティングシステム、アプリケーションプログラム、他のプログラムモジュール、および、プログラムデータを記憶しても良い。

#### 【 0 1 2 8 】

ユーザは、入力装置 2 4 4 0 を介してコンピュータ 2 4 1 0 にコマンドおよび情報を入力できる。入力装置 2 4 4 0 として、非限定的に、キーボード、キーパッド、ポインティ

50

ングデバイス、マウス、タッチペン、タッチパッド、タッチスクリーン、トラックボール、モーションディテクタ、カメラ、マイクロフォン、ジョイスティック、ゲームパッド、スキャナー、もしくは、ユーザがコンピュータと相互作用できる任意の装置を含むことができる。モニタまたは他のタイプの表示装置も、出力インターフェース 2 4 5 0 のようなインターフェースを介してシステムバス 2 4 2 2 に接続される。モニタに加えて、コンピュータは、出力インターフェース 2 4 5 0 を介して接続されるスピーカおよびプリンタのような他の周辺出力装置を含むことができる。1 以上の実施形態において、入力装置 2 4 4 0 は、インターフェース部 2 1 2 にユーザ入力を与え、出力インターフェース 2 4 5 0 は、インターフェース部 2 1 2 から慣性推定器 2 0 2 の動作に関する情報を受信することができる。

10

#### 【0129】

コンピュータ 2 4 1 0 は、リモートコンピュータ 2 4 7 0 のような 1 以上の遠く離れた他のコンピュータに論理的に結合されるネットワーク化または分散化された環境で動作しても良い。リモートコンピュータ 2 4 7 0 は、パーソナルコンピュータ、サーバ、ルータ、ネットワークパソコン (PC)、ピアデバイス (peer device) もしくは他の共通ネットワークノード、または、任意の他のリモートメディアデバイスもしくはリモートメディア伝送デバイスであっても良いし、コンピュータ 2 4 1 0 に関連して説明した要素のいずれか、もしくは、全てを含んでも良い。図 2 4 に表す論理結合は、LAN (local area network) もしくは WAN (wide area network) のような、ネットワーク 2 4 7 2 を含むが、例えば、セルラーネットワークのような、他のネットワーク/バスも含まれる。

20

#### 【0130】

上記のように、例示的な実施形態が、いくつかの演算装置およびネットワーク構造に関連して説明されたが、その背景となる概念は、制約の少ない方式で、メディアを出力または利用することが望まれる任意のネットワークシステムおよび任意の演算装置もしくは演算システムに適用できる。

#### 【0131】

また、同じまたは類似する機能性を実現する多くの方法がある。例えば、適当な API (Application Programming Interface)、ツールキット、ドライバコード、オペレーティングシステム、制御ソフトウェア、独立もしくはダウンロードできるソフトウェアなどにより、ここで説明する技術を利用したアプリケーションおよびサービスが可能となる。このように、実施形態では、ソフトウェアまたはハードウェアと同じように、API (もしくは、他の同じ機能のソフトウェア) の観点から、ここに説明した 1 以上の形態を実行することが期待される。そして、ここに説明するいくつかの実施形態は、全てがハードウェア、一部がハードウェア、および、全てがソフトウェア、一部がソフトウェアの態様であっても良い。

30

#### 【0132】

ここで「例示的」という言葉は、例または模式化を意味するために用いられている。疑義を避けるために、ここに開示される態様は、そのような例に限定されない。さらに、例として、ここに説明する任意の態様またはデザインは、他の態様もしくはデザインよりも好ましい、または、有利であると解釈される必要はなく、また、等価な構造および技術を除外するものでもない。さらに、疑義を避けるために「含む」、「有する」および他の類似する語を、詳細な説明、または、請求の範囲のいずれかに使用する限りにおいて、そのような語は、「備える」と類似した用法であって、任意の付加もしくは他の要素を排除しない。

40

#### 【0133】

演算装置は、通常、コンピュータが読み取り可能な記憶媒体 (例えば、メモリ 2 1 6 もしくは 1 1 1 2) および通信媒体、もしくは、そのいずれかを含むいくつかの媒体を備える。ここで用いられる、これらの 2 つの語は、次のように相互に異なる。コンピュータが読み取り可能な媒体は、任意の利用可能な媒体であって、コンピュータがアクセス可能で有り、通常、一過性でない性質を有し、揮発性および不揮発性の両方のものと、脱着可能

50



および脱着不可能の両方のものを含む。非限定的な例として、コンピュータが読み取り可能な記憶媒体は、コンピュータが読み取り可能な命令、プログラムモジュール、構造的データもしくは非構造的データのような情報を保存するために、任意の方法もしくは技術に関連して実装される。コンピュータが読み取り可能な記憶媒体は、RAM、ROM、EEPROM、フラッシュメモリもしくは他のメモリ技術、CD-ROM、DVD (digital versatile disk) もしくは他の光学的記録ディスク、磁気カセット、磁気テープ、磁気記憶ディスクもしくは他の磁気記憶デバイス、望ましい情報を保存するために使用される他の具体的および非一過性の媒体、もしくはそのいずれかを含むが、これらに限定される訳ではない。コンピュータが読み取り可能な記憶媒体は、それに保存された情報に関するいくつかの動作のために、1以上のローカルもしくはリモートの演算装置により、例えば、アクセス要求、検索要求または他のデータ検索プロトコルを介してアクセスされる。

10

#### 【0134】

一方、通信媒体は、通常、コンピュータが読み取り可能な命令、データ構造、プログラムモジュールもしくは他の構造データまたは非構造データを、例えば、搬送波もしくは他の輸送手段である変調データ信号のようなデータ信号に一体化し、任意の情報配信または輸送媒体を含む。「変調データ信号」または信号という語は、1以上の信号に情報を符号化する方法により設定もしくは変更された1以上の特性を有する信号に言及する。非限定的な例によれば、通信媒体は、有線ネットワークもしくは直接配線の結合のような有線媒体、および、音波、高周波、赤外および他の無線媒体のような無線媒体を含む。

#### 【0135】

20

前述したように、ここに説明されるいくつかの技術は、ハードウェアまたはソフトウェア、もしくは、適宜、その結合に関連して実施される。ここに使用する「要素」、「システム」などの用語は、さらに、ハードウェア、ハードウェアとソフトウェアの結合、ソフトウェア、もしくは、ソフトウェアの実行のいずれかであるコンピュータに関連する要素に言及することを意図している。例えば、要素は、プロセッサで実行される処理、プロセッサ、オブジェクト、実行ファイル、実行スレッド、プログラムおよびコンピュータ、もしくは、そのいずれかであっても良いが、これらに限定される訳ではない。例として、コンピュータ上で実行されるアプリケーションおよびコンピュータの両方が、要素であっても良い。1以上の要素が、プロセスおよび実行スレッド、もしくは、そのいずれかの中に存在し、要素は、1つのコンピュータに局在しても良いし、2以上のコンピュータに分散されても良い。さらに、「デバイス」は、特別に設計されたハードウェア、また、ハードウェアに特定の機能（例えば、符号化および複合化、もしくはそのいずれか）を実施させるソフトウェアの実行により特化される一般のハードウェアとしてもたらされても良い。

30

#### 【0136】

前述したシステムは、いくつかの要素の相互作用に関連して説明される。そのようなシステムおよび要素は、これらの要素もしくは特定された準要素、特定された要素または準要素のいくつか、および、付加的な要素、もしくは、そのいずれかを含み、前述のいくつかの置き換えおよび結合にしたがうことを認識すべきである。準要素は、親要素の内部（階層構造）に含まれるのではなく、他の要素と通信可能に結合された要素として実装されても良い。さらに、1以上の要素は、集約された機能性を与える単一の要素に結合されても良いし、いくつかの離間した準要素に分割されても良いこと、また、マネジメント層のような任意の1以上の中間層が、集積化された機能性を与えるために、そのような準要素に通信可能に結合されても良いことに留意すべきである。ここに説明する任意の要素は、ここに具体的に説明されないが、当業者に一般的に知られた1以上の他の要素と相互に作用しても良い。

40

#### 【0137】

ここに説明した多くのインターフェース（例えば、音響セグメント）を提供または役立たせるために、ここに説明する要素は、アクセスが承認されたデータの全体もしくはそのサブセットを試験し、事象およびデータ、もしくはそのいずれかを介して捉えられた観察結果から、システム、環境などの状態を論証もしくは推定することができる。推定は、具

50

体的な背景または動作を決定するために用いられ、例えば、状態の確率分布を生成することができる。推定は確率的、すなわち、データおよび事象の考察に基づいて対象の状態の確率分布の計算であっても良い。推定は、事象およびデータ、もしくは、そのいずれかから、より高度の事象を構成するために用いる技術に言及することもできる。

#### 【0138】

そのような推定は、観測された事象および保存された事象のデータのセット、および、事象が一時的な近接状態において相関するかどうか、また、事象およびデータが、1またはいくつかの事象およびデータソースから生じたかどうか、から新規の事象または動作を構成することができる。いくつかの(明示的および黙示的に備えられた、もしくは、そのいずれかである)分類スキームおよびシステム、もしくは、そのいずれか(例えば、サポートベクトルマシン(support vector machine)、ニューラルネットワーク(neural network)、エキスパートシステム(expert system)、簡易ベースネットワーク(Bayesian belief network)、ファジーロジック(fuzzy logic)、データフュージョンエンジン(data fusion engine)など)は、クレームされた主題に関連した自動化および推定された動作を実施に用いることができる。

10

#### 【0139】

分類器は、入力の属性ベクトル、 $x = (x_1, x_2, x_3, x_4 \cdots x_n)$ を、関数 $f(x)$ によりマップし、その入力が入力の1つのクラスに属することを確定させる。そのような分類には、ユーザが自動的に実行されることを望む動作を予知もしくは推定するために、確率的小および統計に基づいた解析(例えば、解析設備および解析コストへの組み込み)を用いる。サポートベクトルマシン(support vector machine: SVM)は、用いることが可能な分類器の1例である。SVMは、可能な入力空間におけるハイパーサーフェース(hyper-surface)を見つけることにより動作する。ここで、ハイパーサーフェースは、非誘因事象からの誘因基準を分割する。直感的に、これは、訓練データに近いが同一でないテストデータに対して、分類を修正する。分類のための他の直接的および間接的なモデルとして、ナイーブベイズ(naive Bayes)、ベイジアンネットワーク(Bayesian network)、ディシジョンツリー(decision tree)、ニューラルネットワーク、ファジーロジックモデル(fuzzy logic model)、および、独立性の異なるパターンを与える確率分類モデルを用いることができる。ここで用いられる分類は、モデルの優先度の解明に用いられる統計的な不具合も内包する。

20

30

#### 【0140】

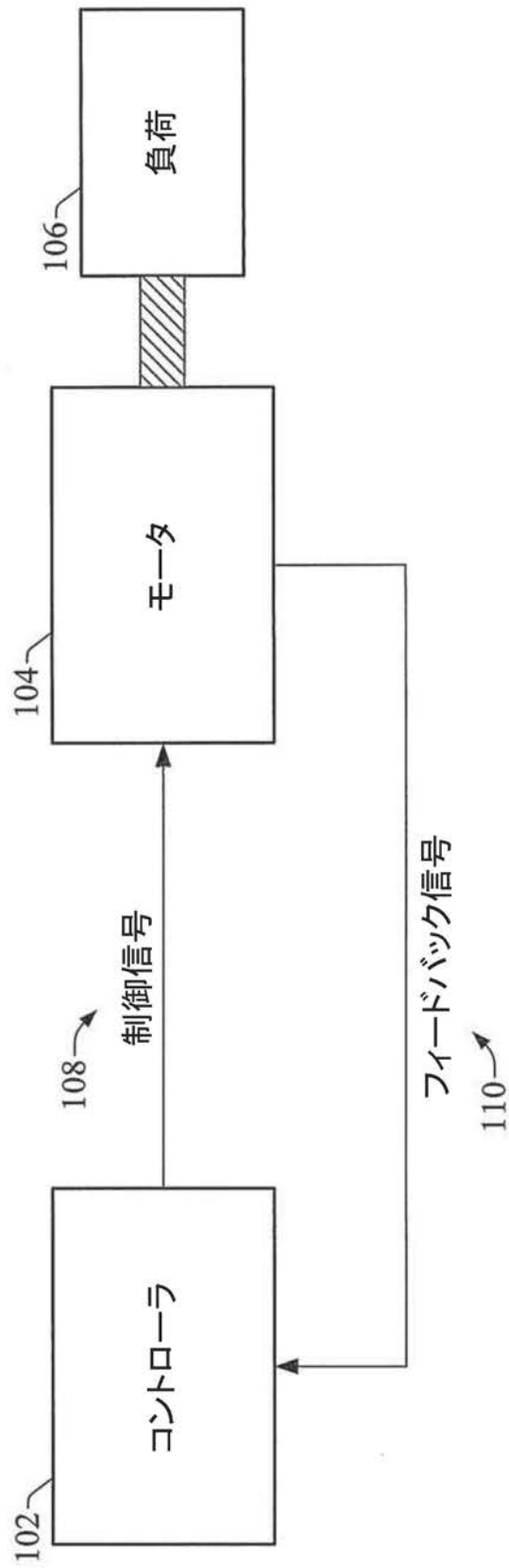
上記の例示的なシステムの視点において、説明された主題にしたがって実行される手順は、いくつかのフローチャート(例えば、図9、10および20~22)参照して良く認識できるであろう。説明の簡略化の観点から、手順は、連続したブロックとして示され、説明されるが、クレームされた主題は、ブロックの順序により限定されることはなく、いくつかのブロックは、ここに表され説明されたものと異なる順序であっても良いし、他のブロックと同時に実行されても良いことを理解し、認識すべきである。逐次的でない分岐したフローがフローチャートに表されているが、いくつかの他の分岐、フローパスおよびブロックの順序が実行され、同じ、または、類似した結果が得られることが認識できる。さらに、以下に説明する手順の実施では、すべてのブロックが必要とされないであろう。

40

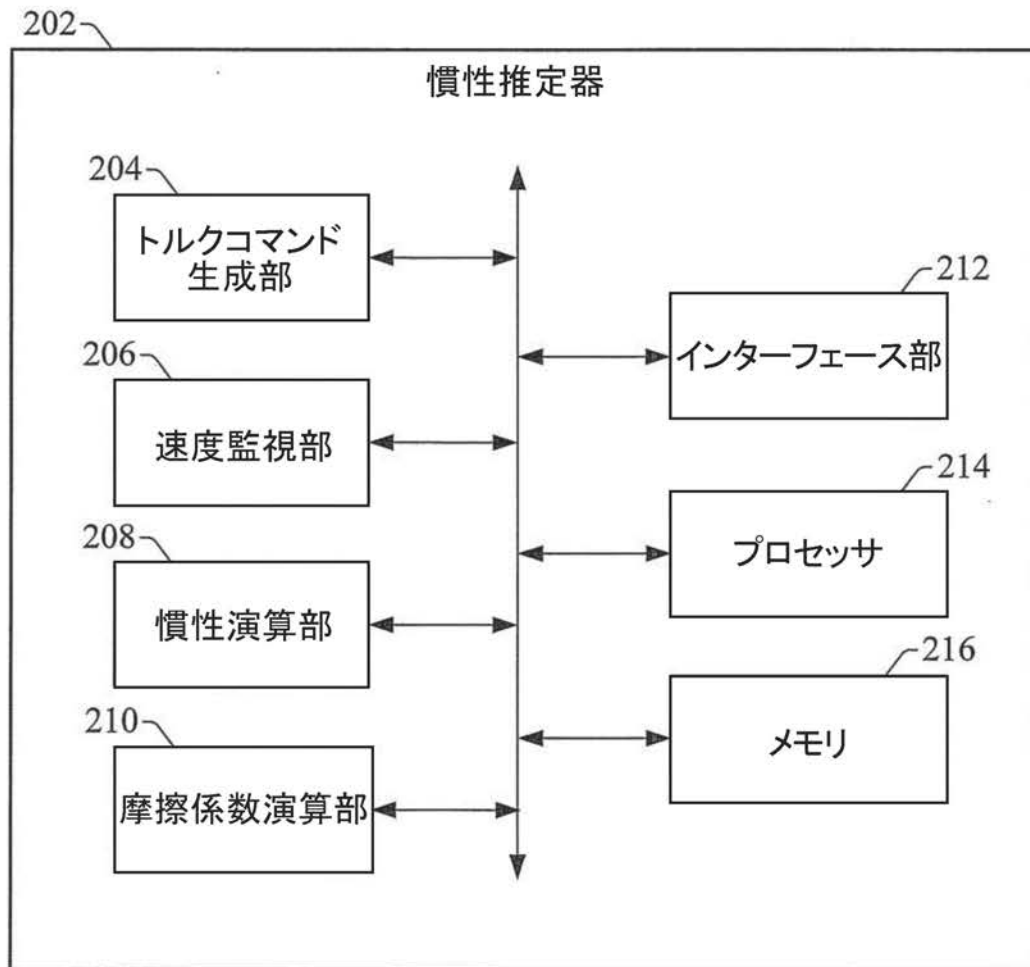
#### 【0141】

ここに説明されたいくつかの実施形態に加えて、他の類似した実施形態を用いることができ、対応する実施形態の同じもしくは類似した機能を実施するために、上記の実施形態から逸脱することなく、その変形および付加が可能である。さらに、多数のプロセスステップまたは多数のデバイスは、ここに説明する1以上の機能を協働して実行することができ、複数のデバイスにまたがった保存も、同じように可能である。したがって、本発明は、いかなる実施形態にも限定されず、添付されたクレームの広さ、精神および範囲を解釈することができる。

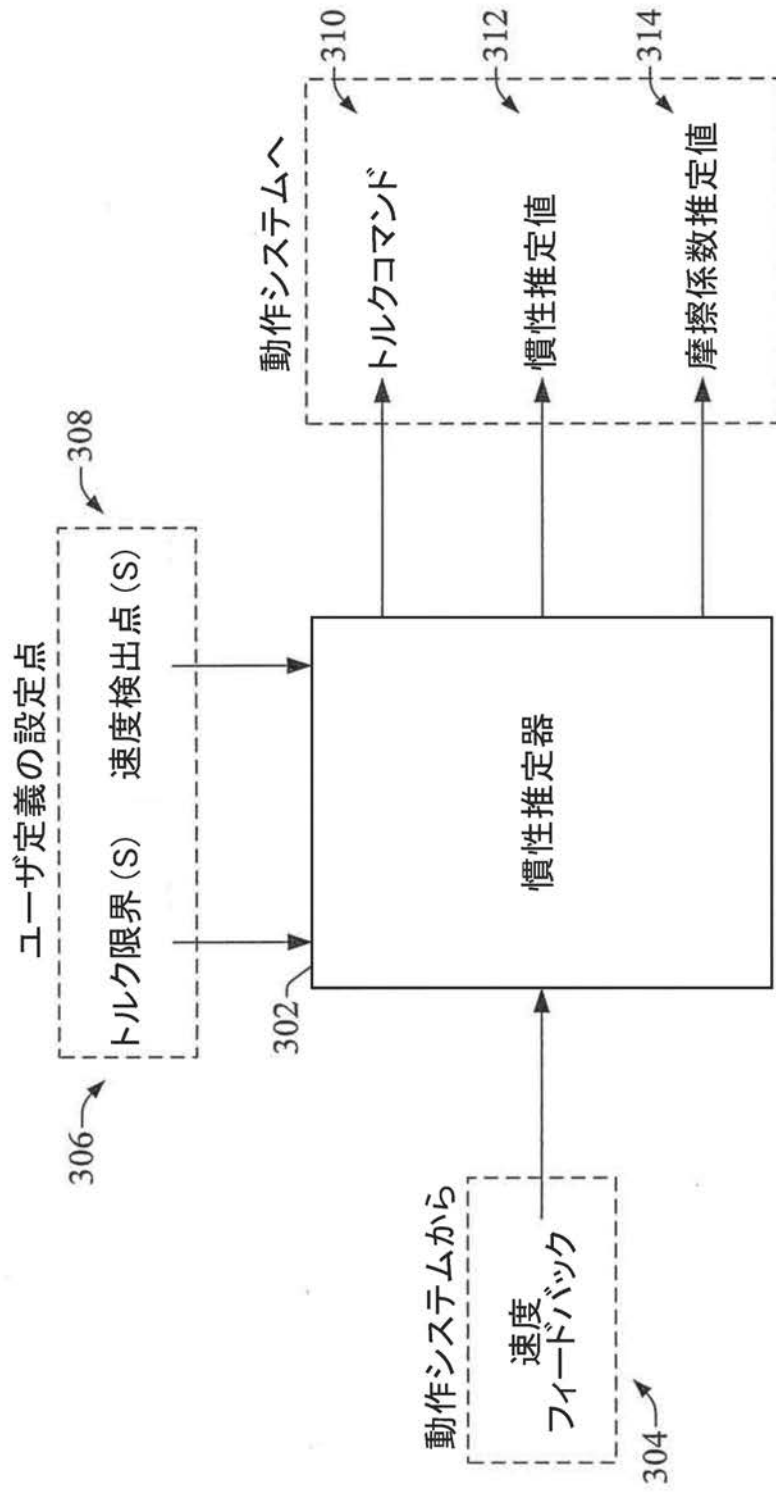
【図 1】



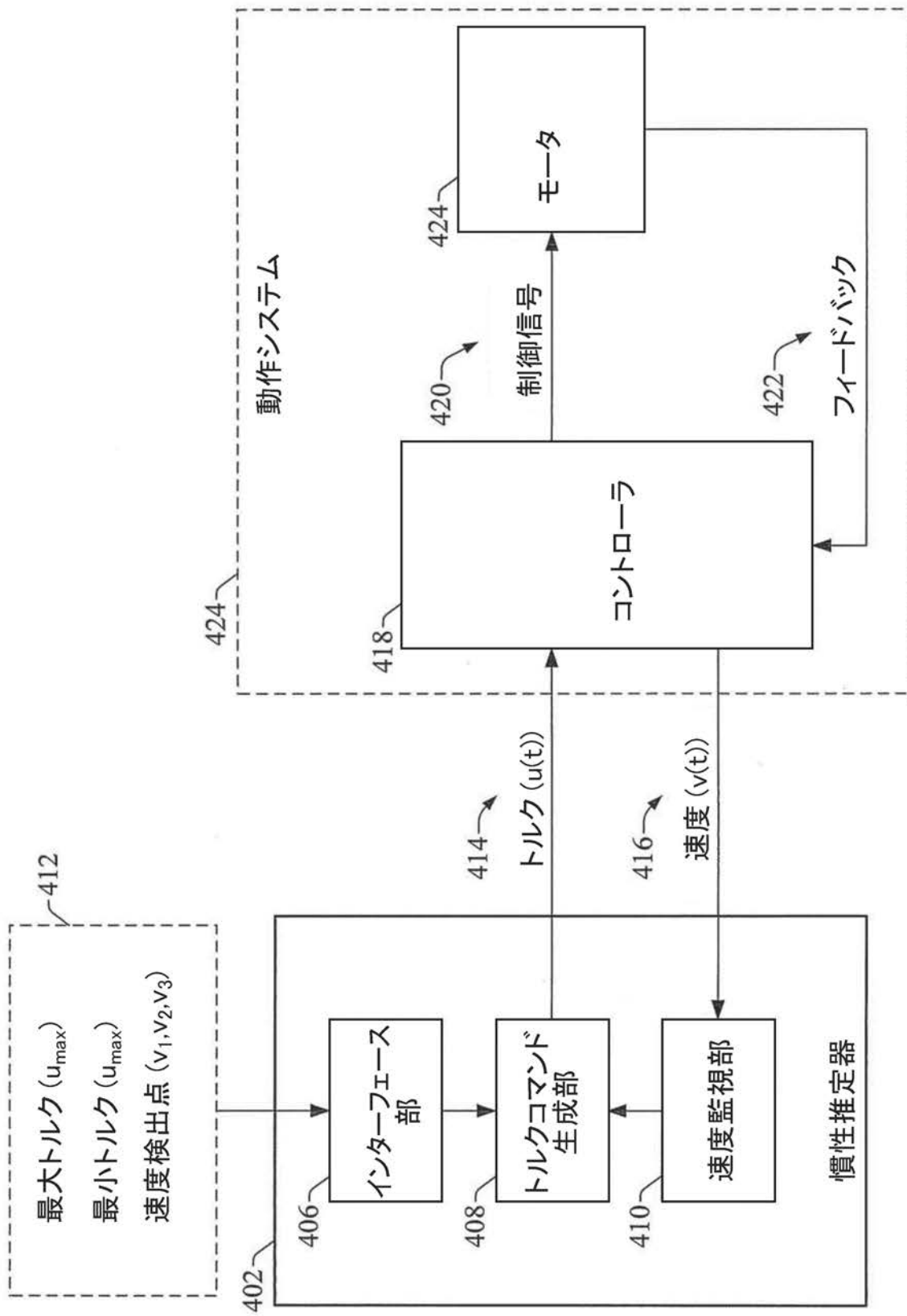
【図 2】



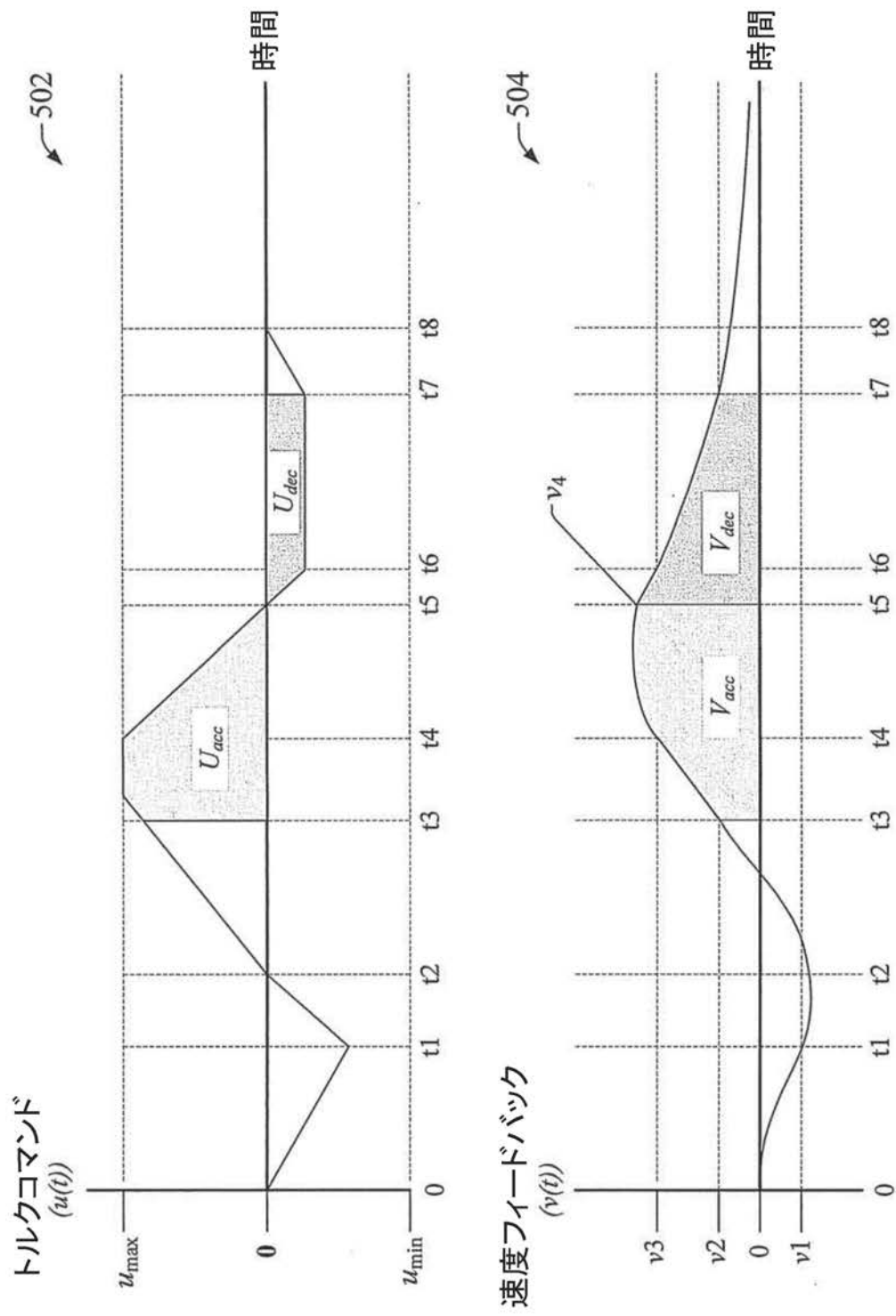
【図 3】



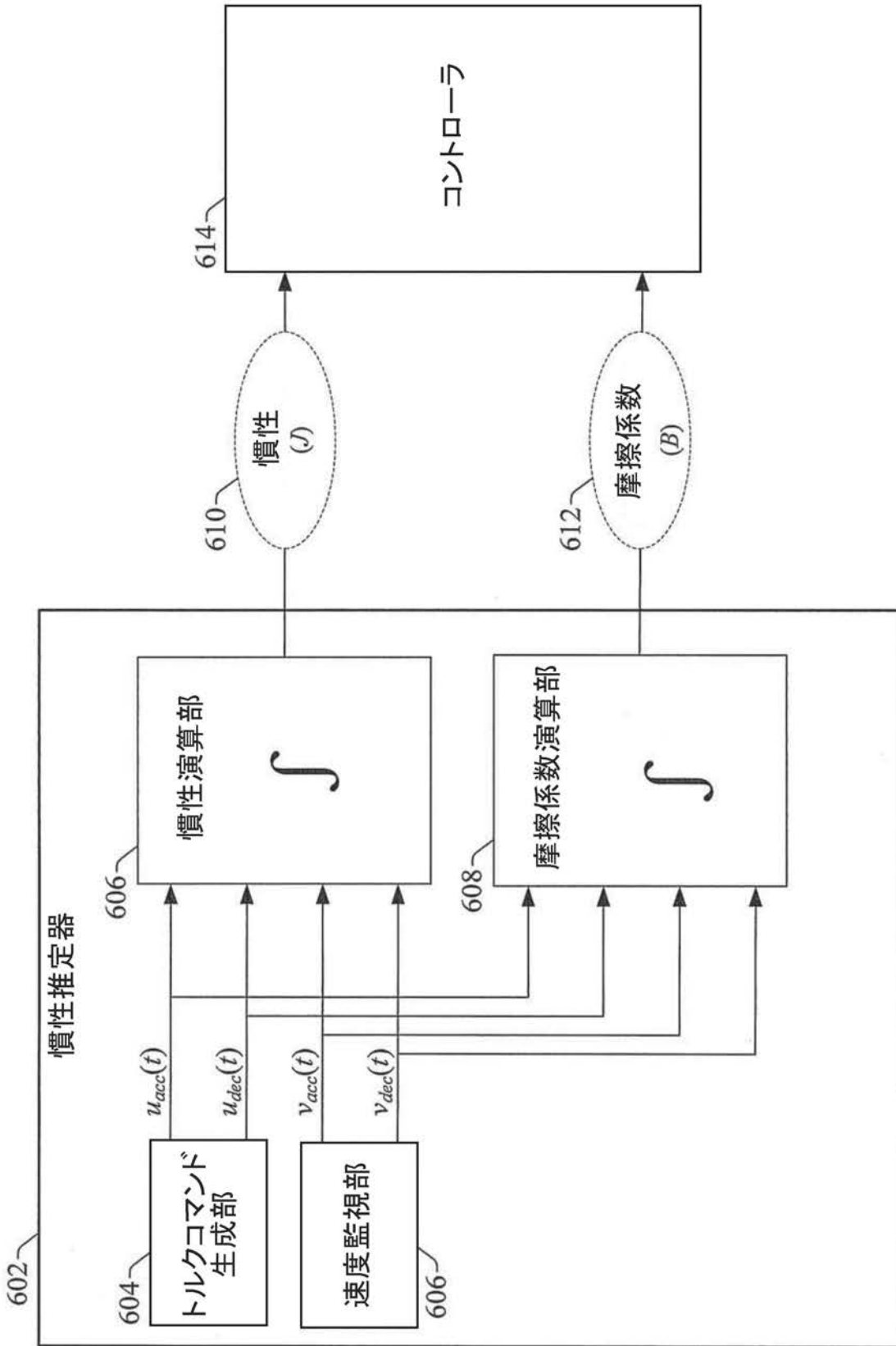
【図 4】



【図 5】

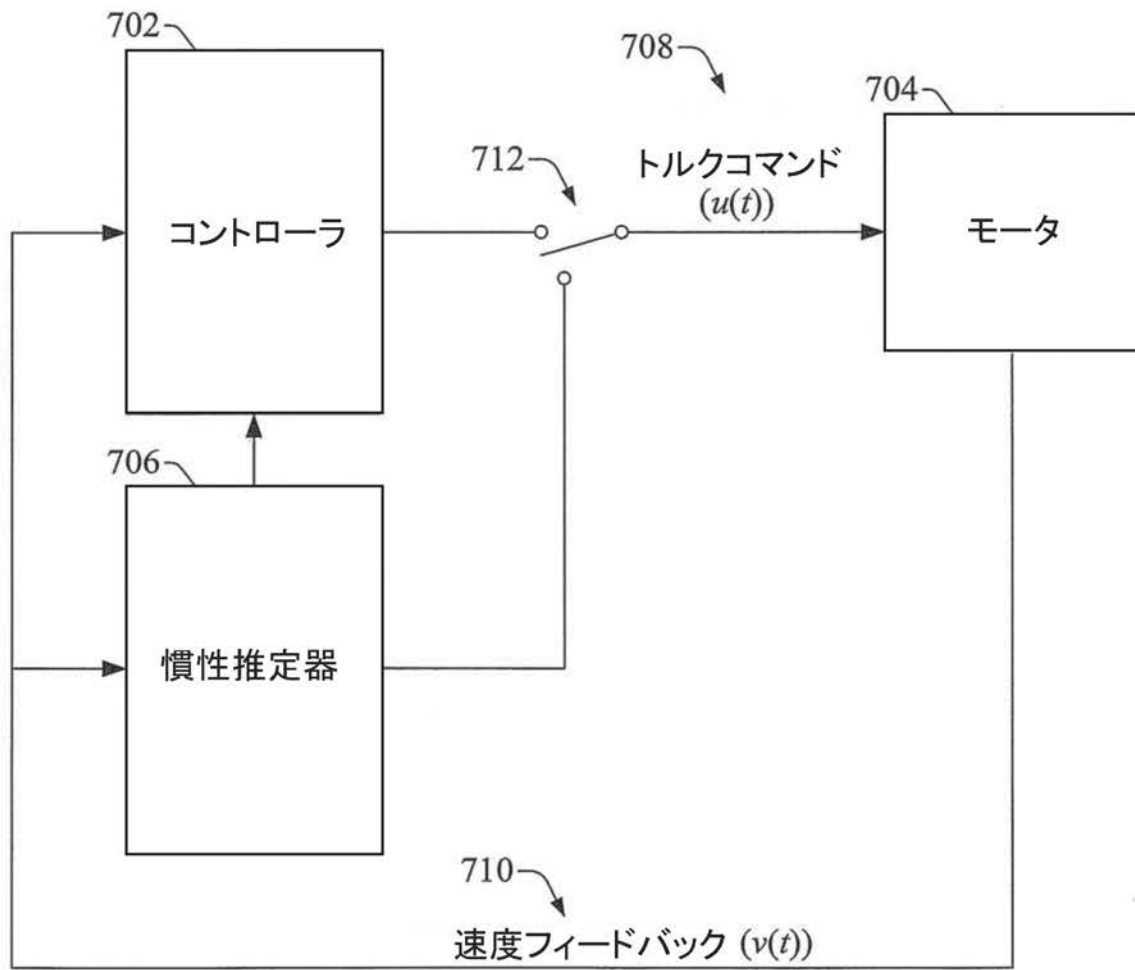


【図 6】

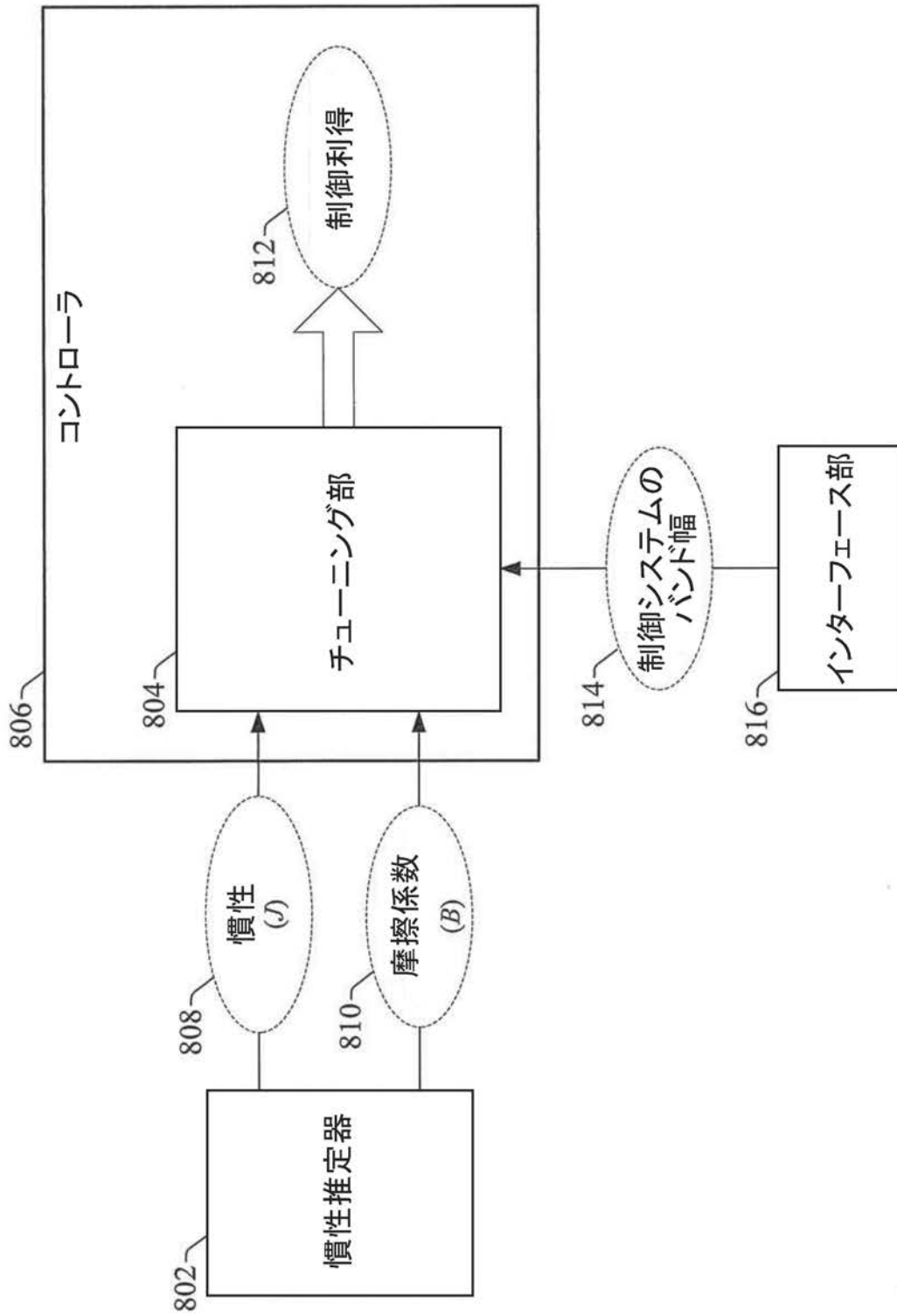




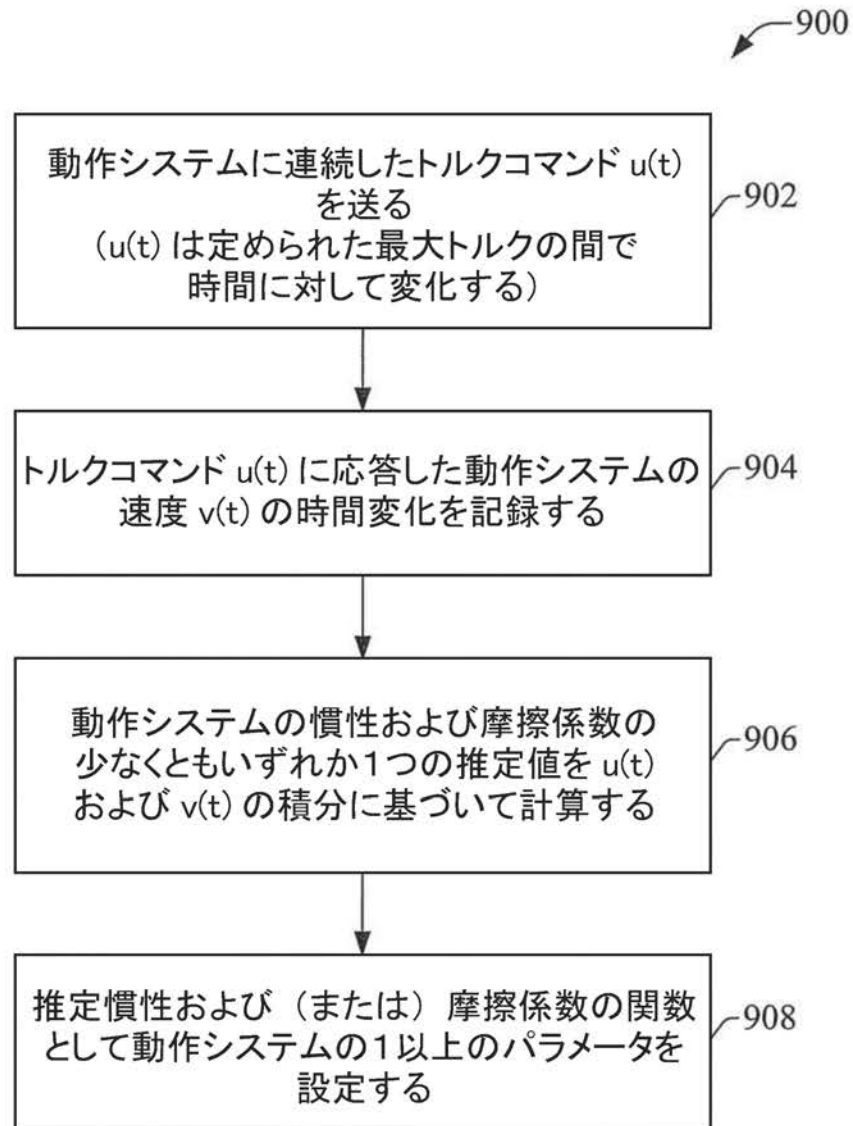
【図 7】



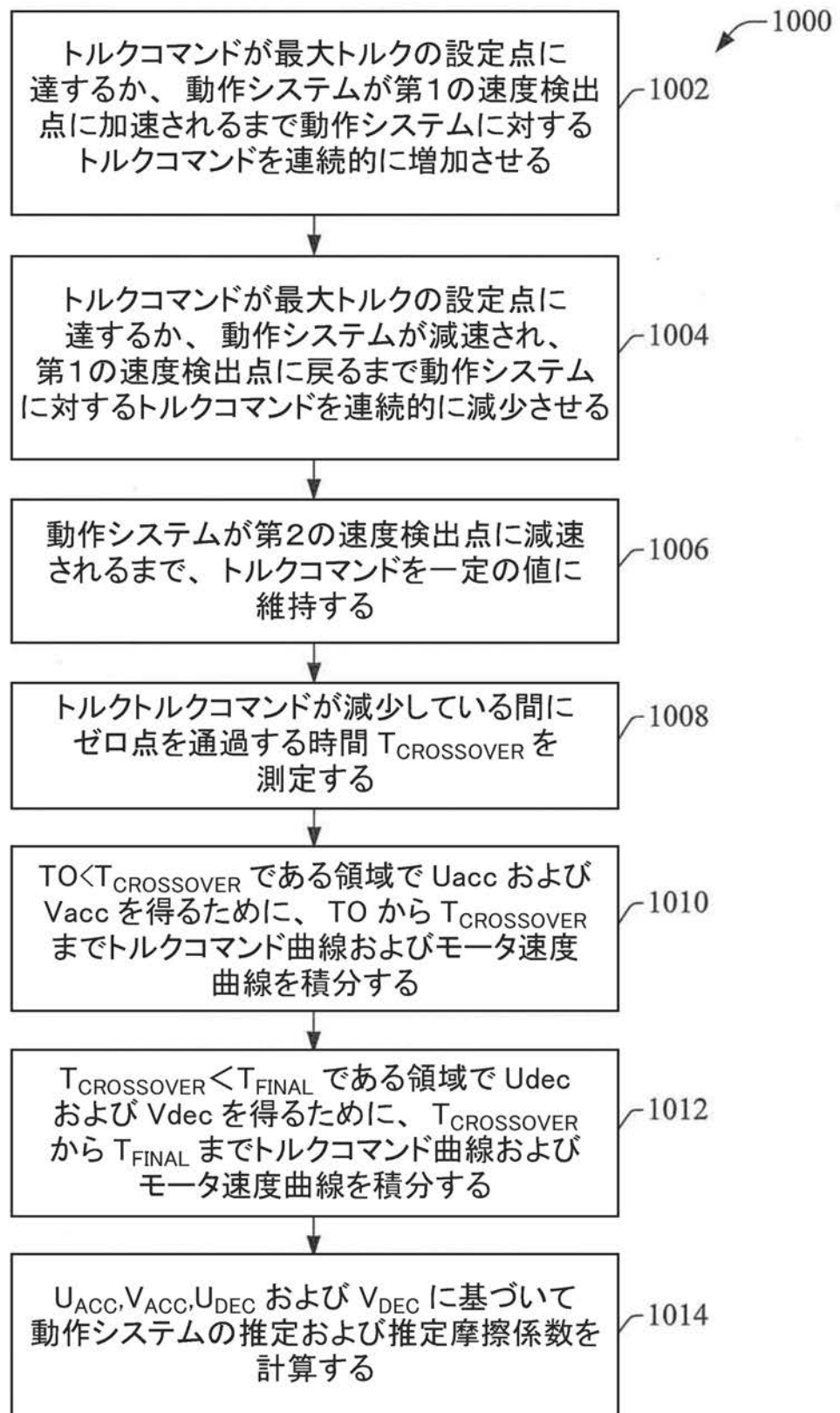
【図 8】



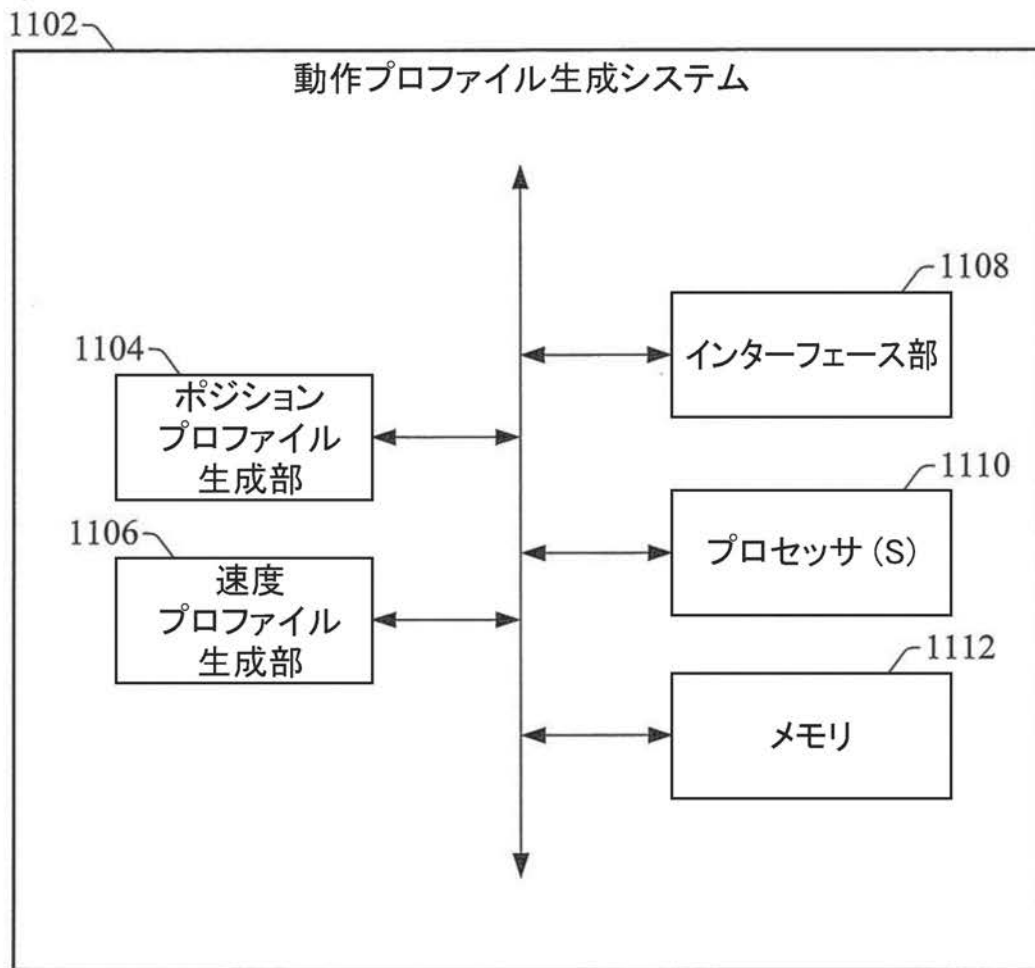
【図 9】



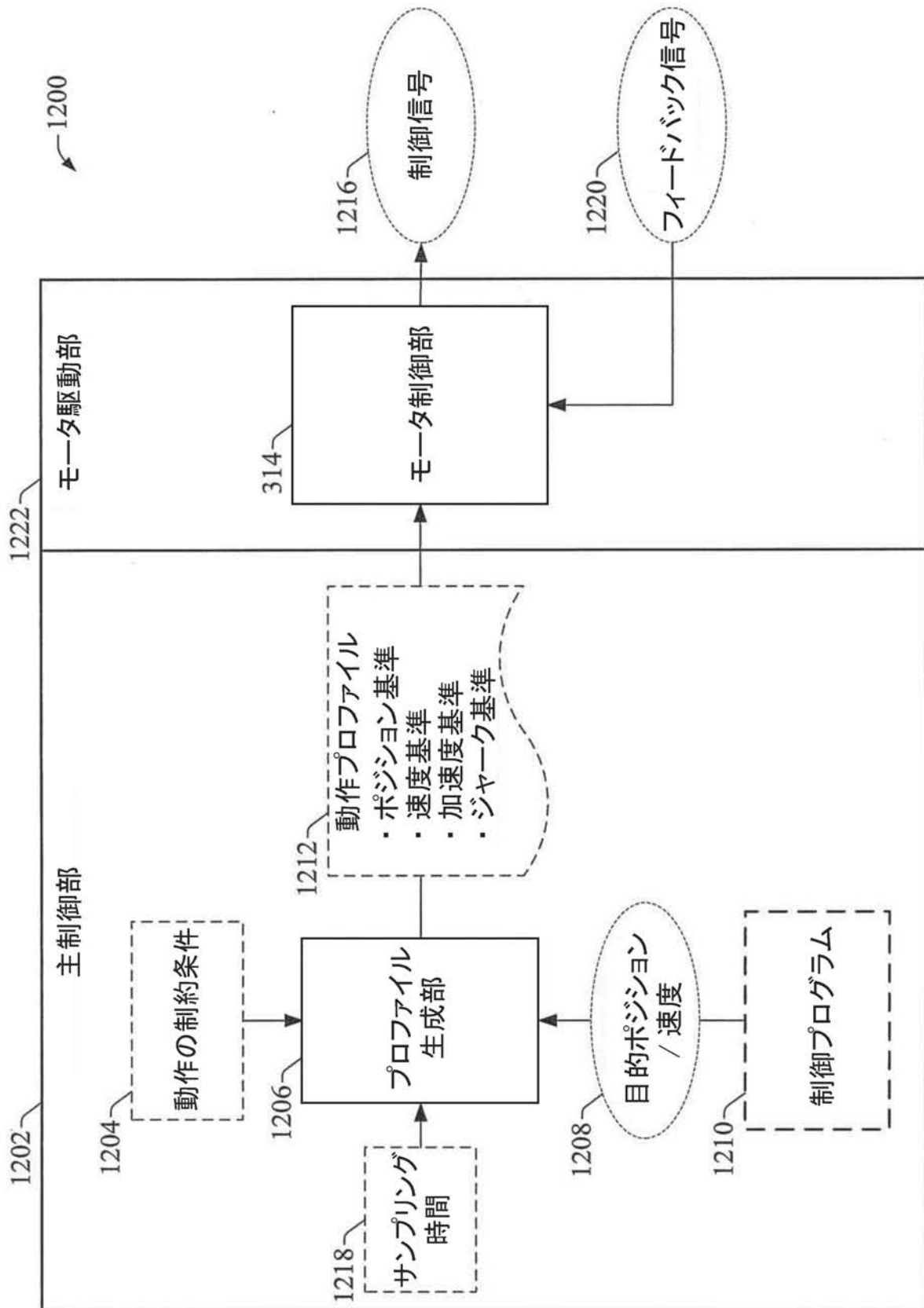
【図 10】



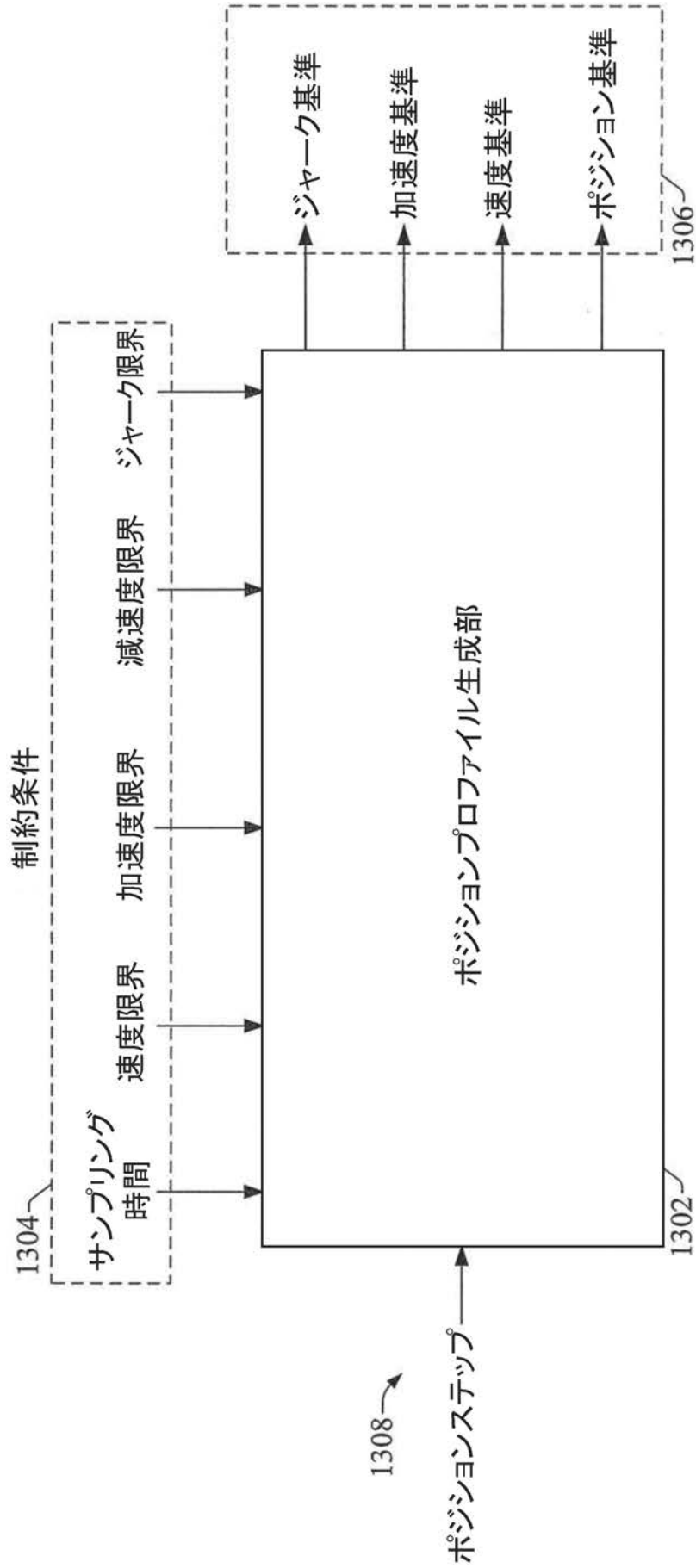
【図 11】



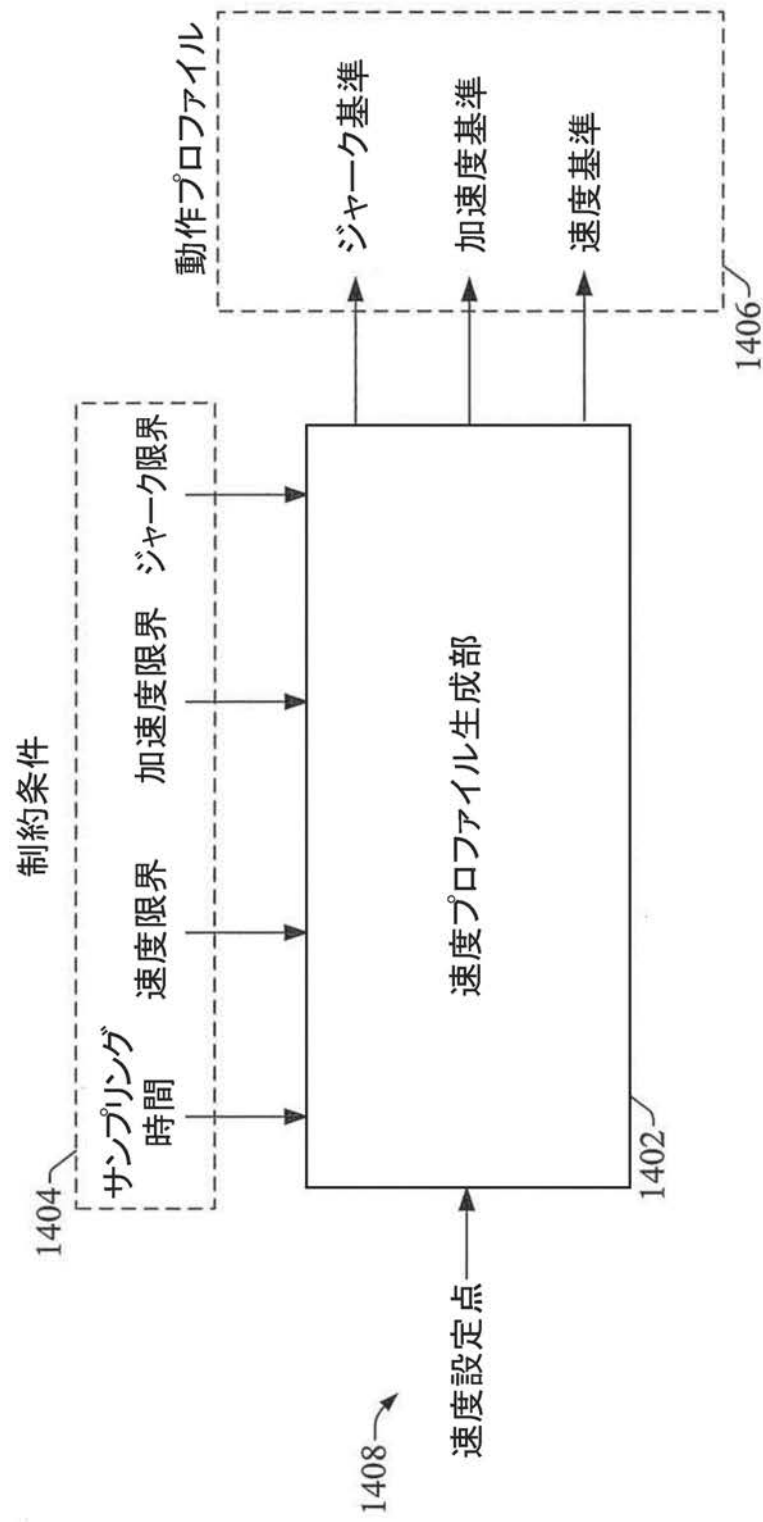
【図 12】



【図 13】

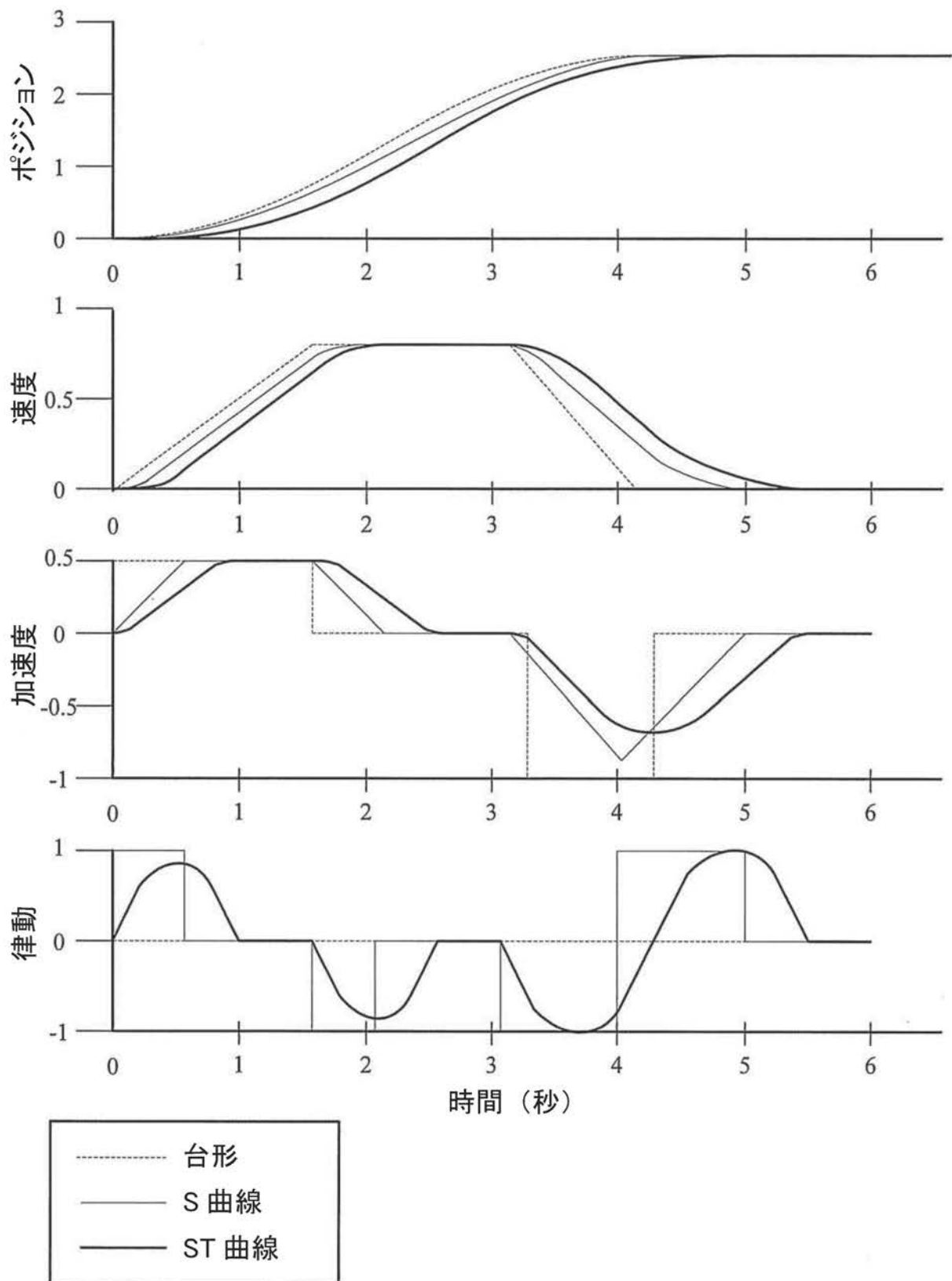


【図 14】

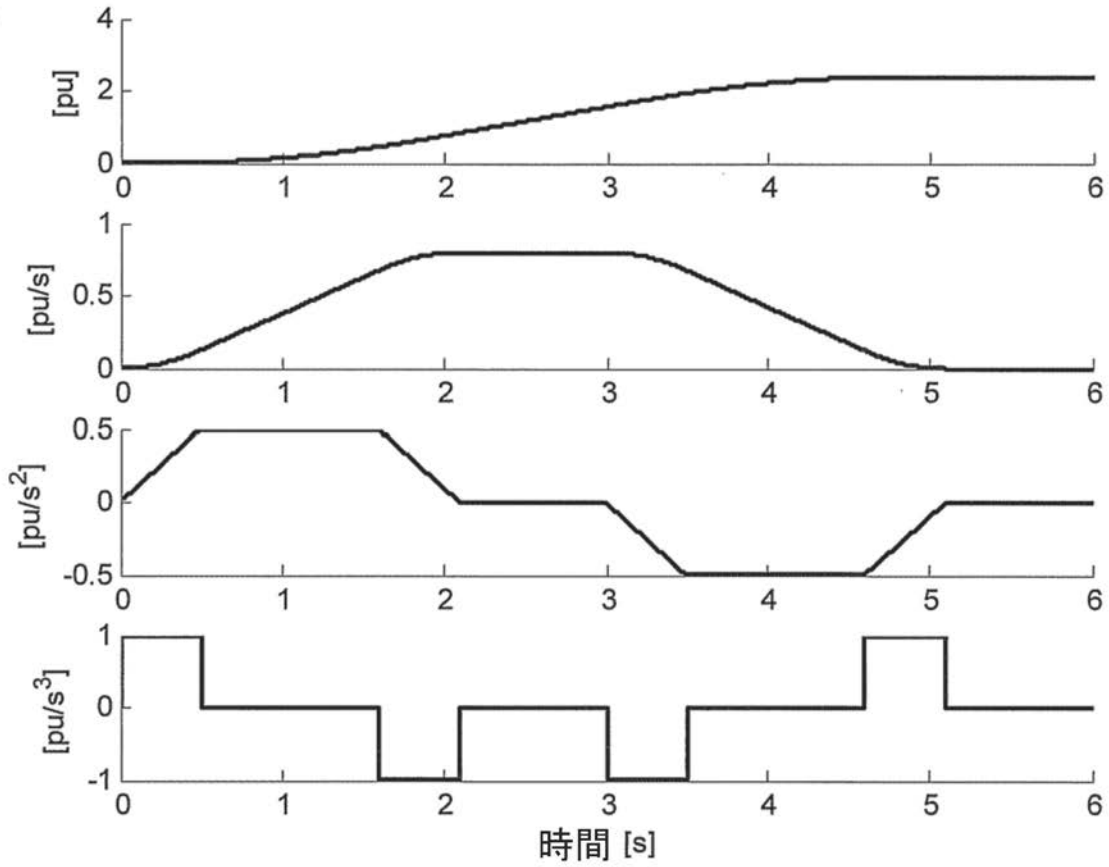




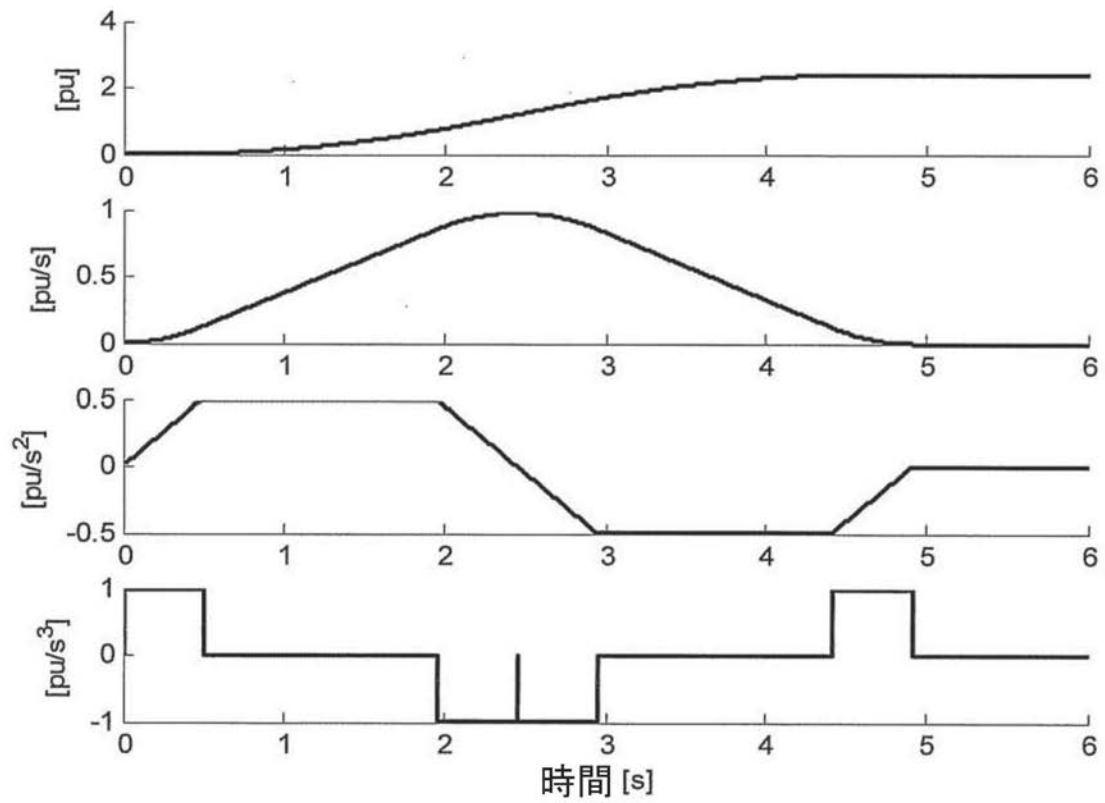
【図 15】



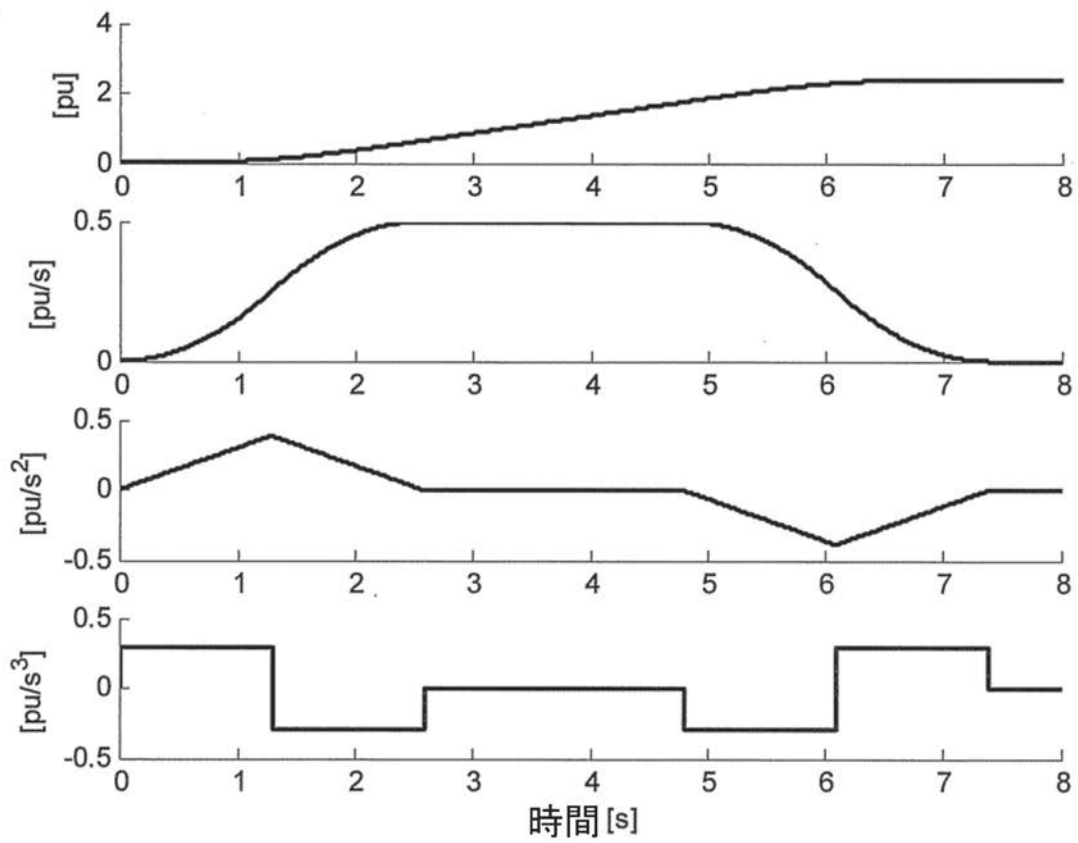
【図 16】



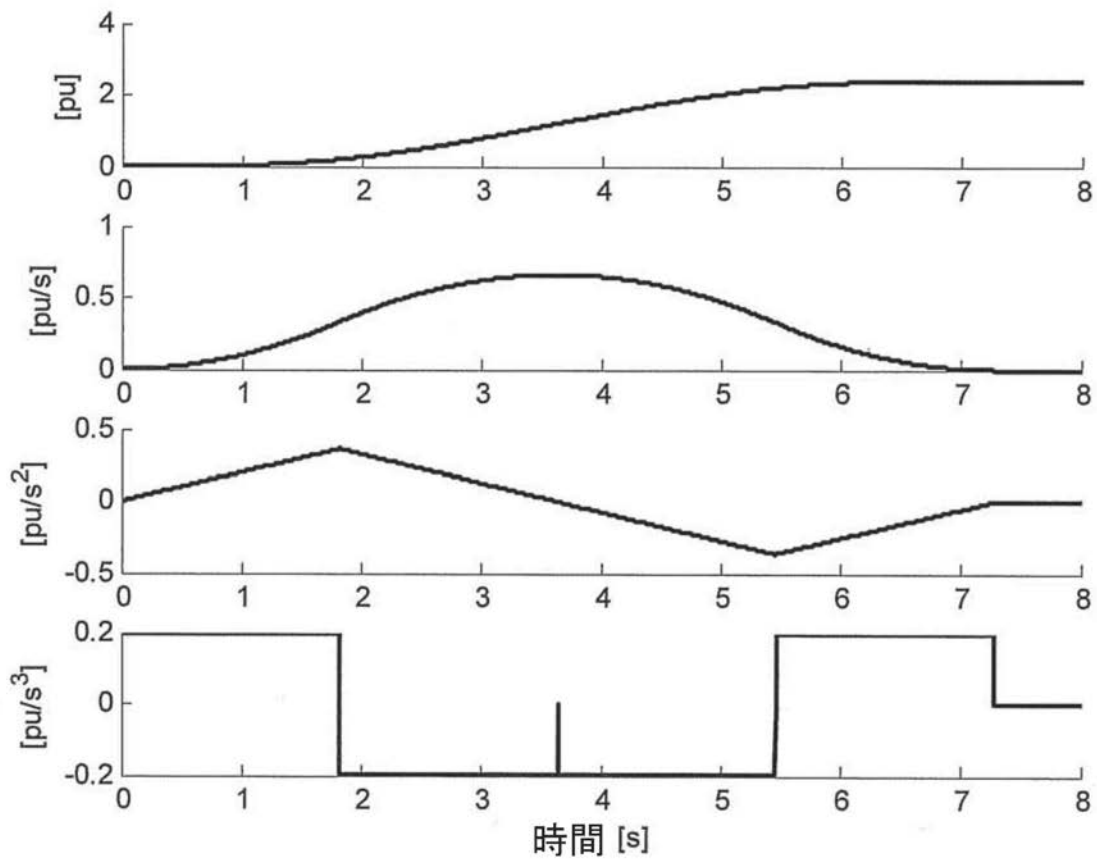
【図 17】



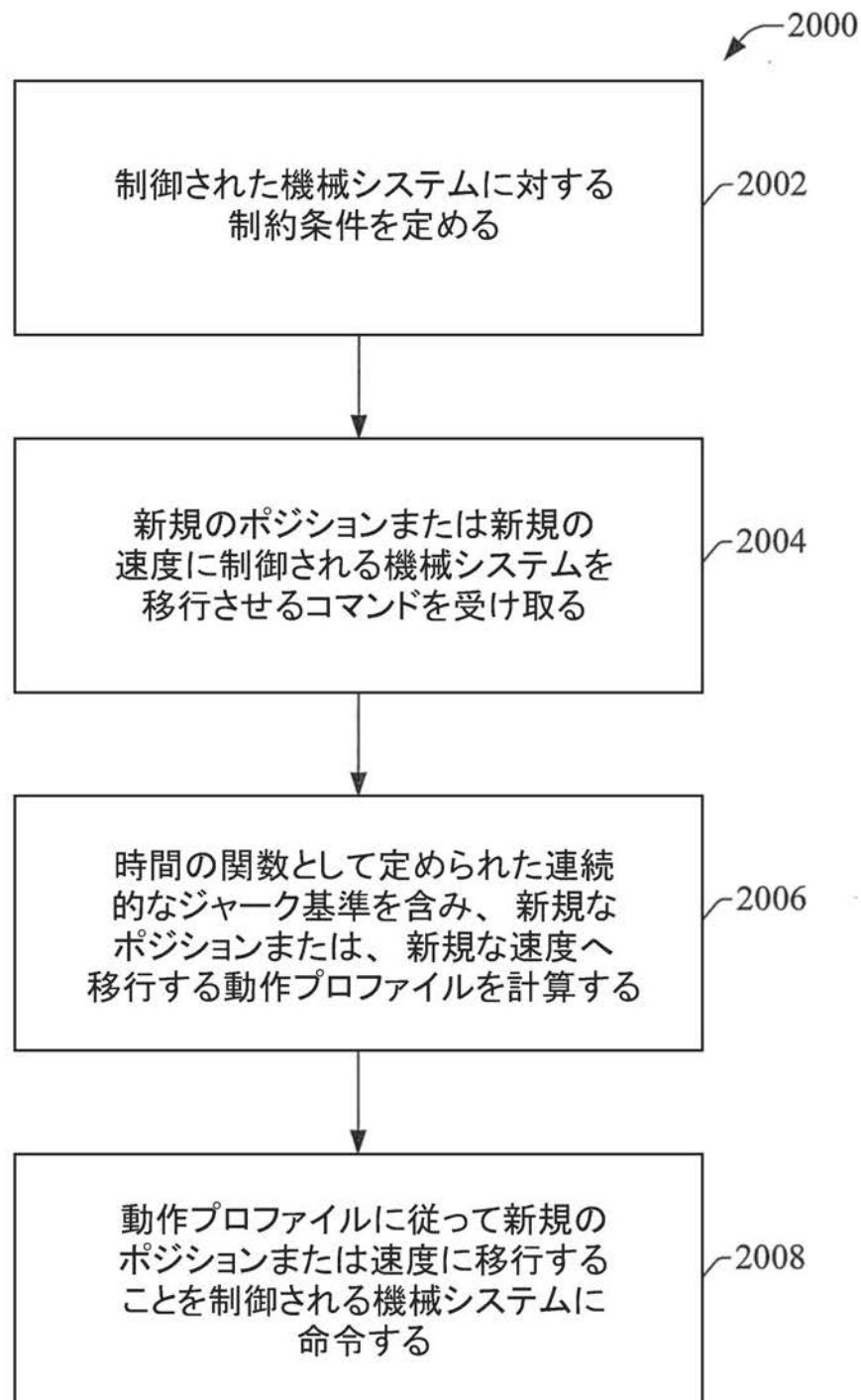
【図 18】



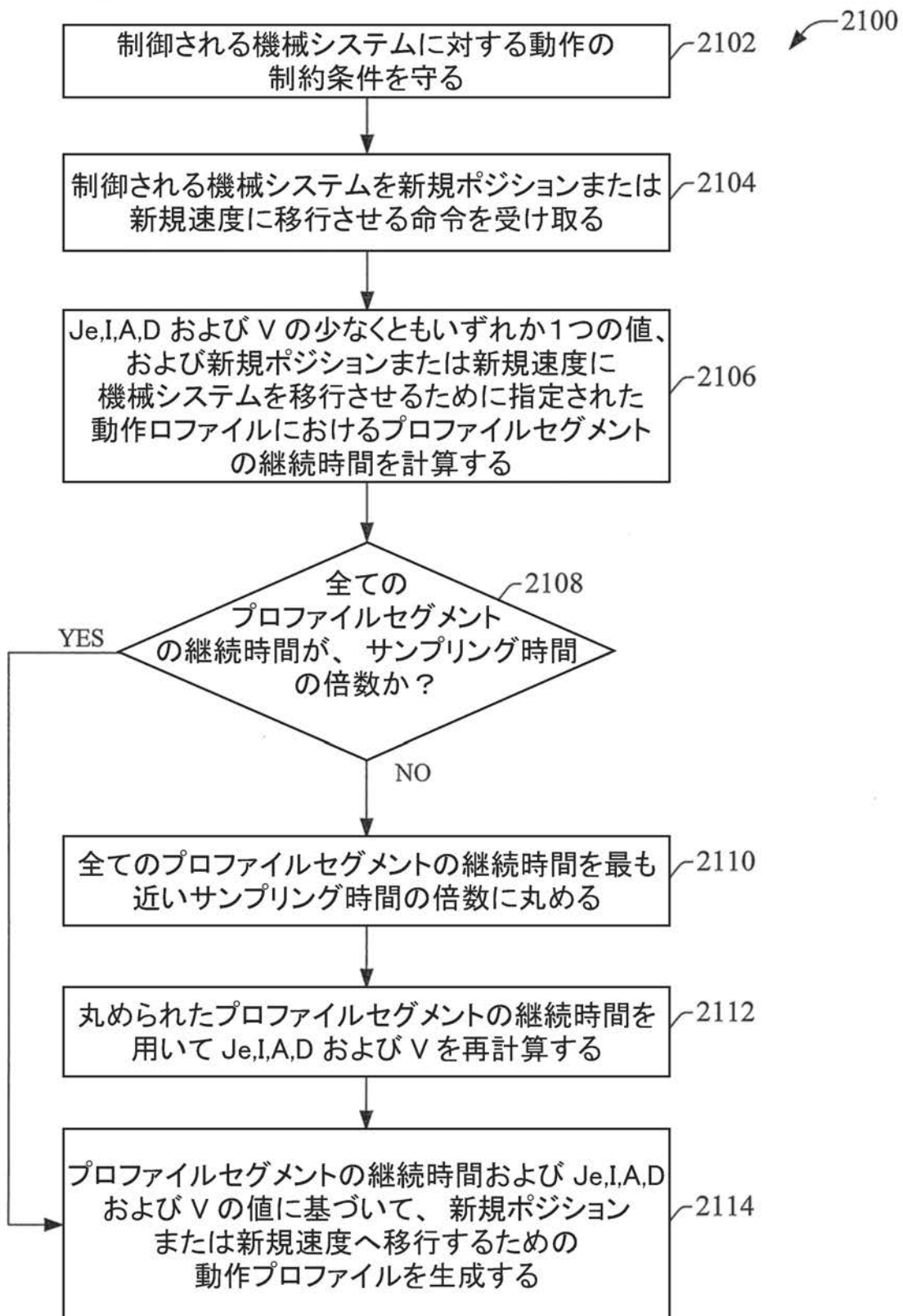
【図 19】



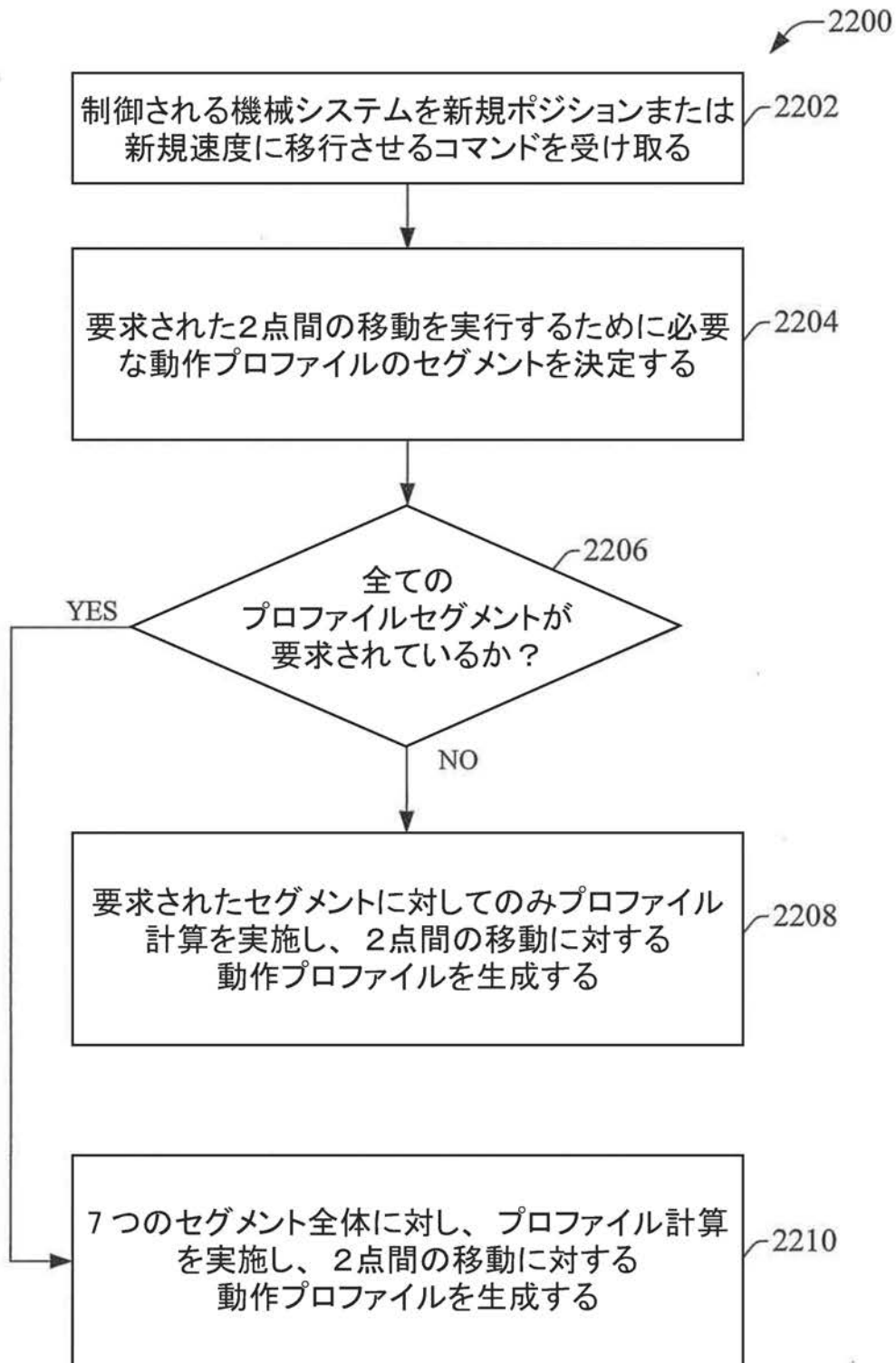
【図 20】



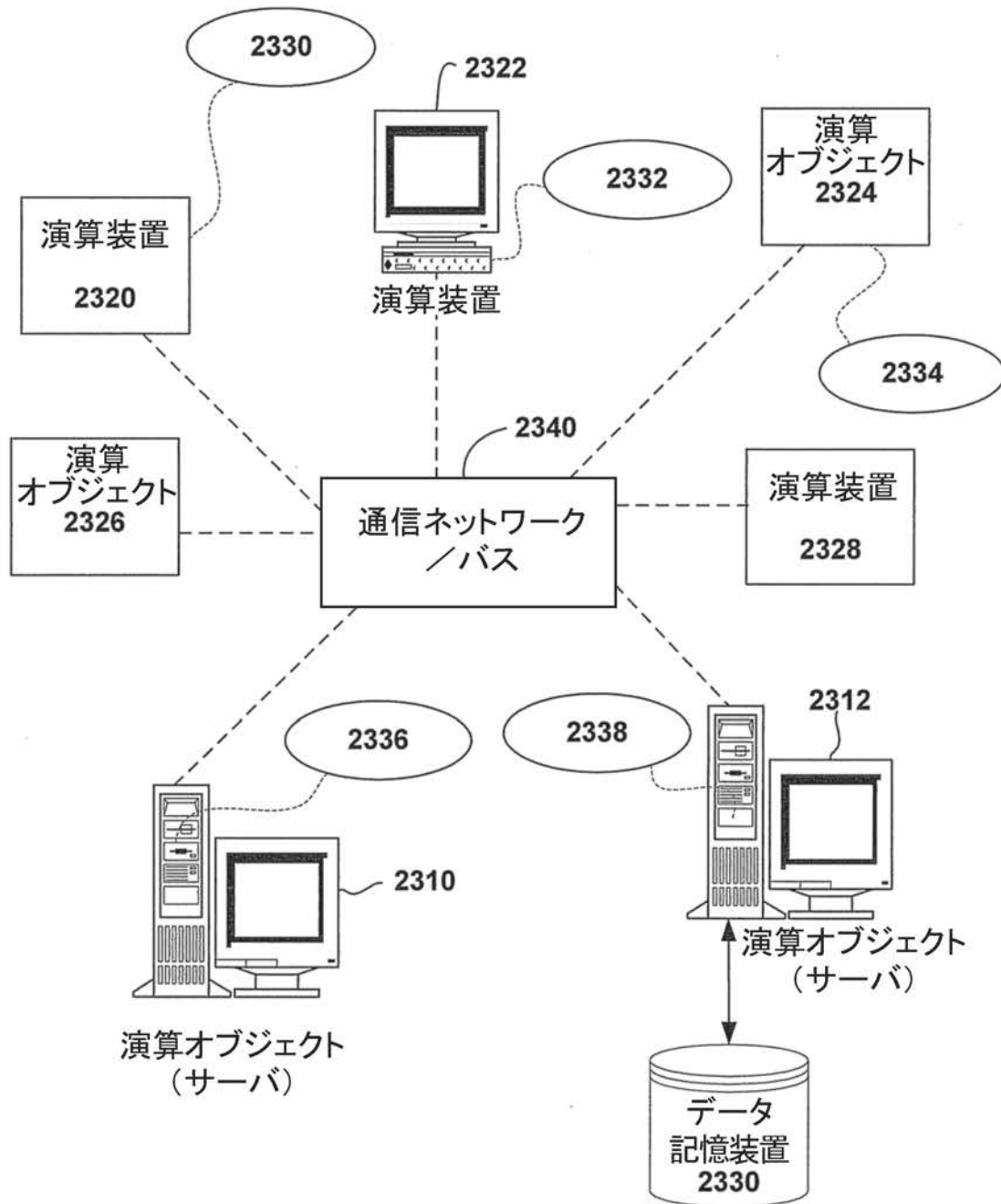
【図 2 1】



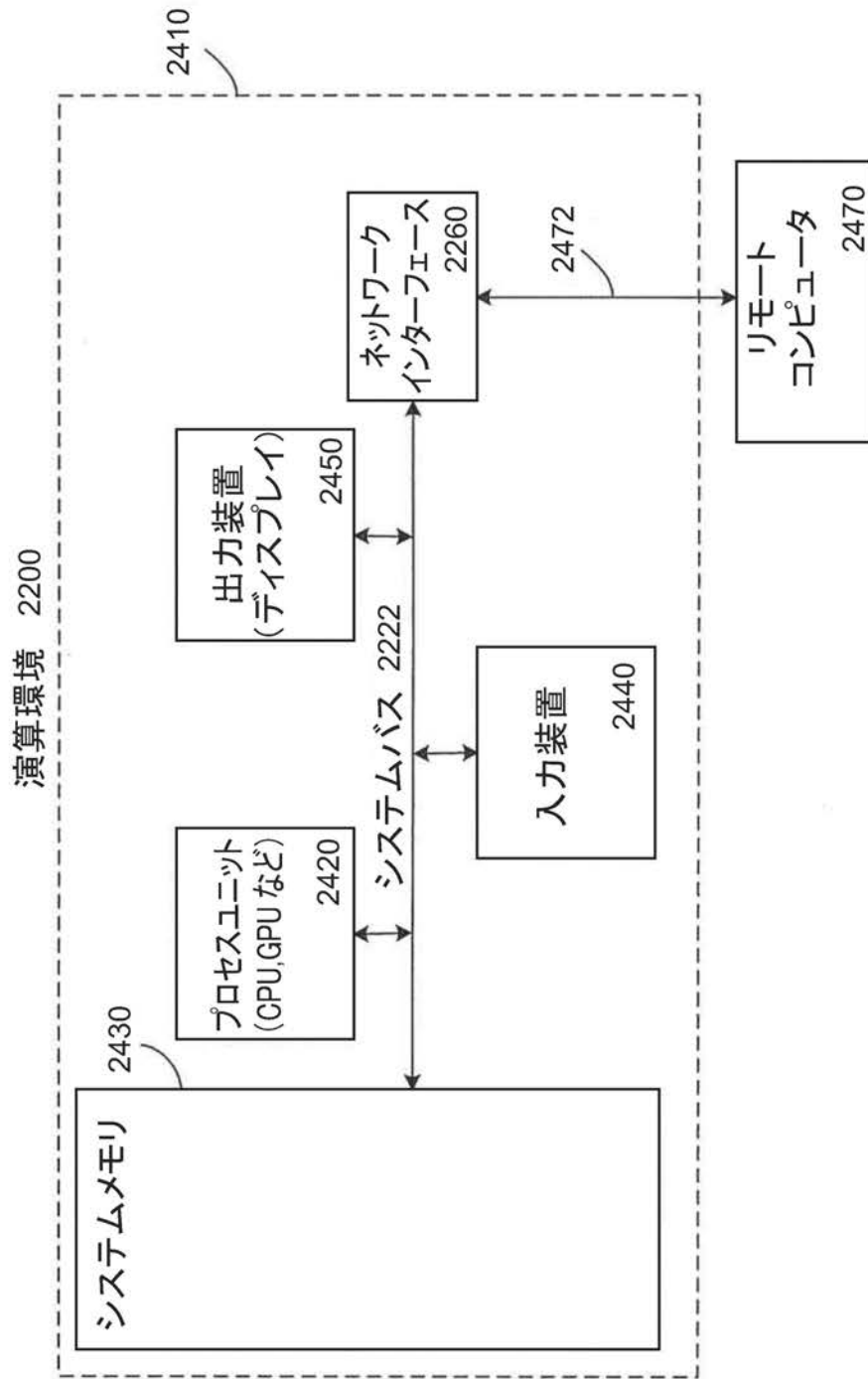
【図 2 2】



【図 23】



【図 24】





---

フロントページの続き

(74)代理人 100157901

弁理士 白井 達哲

(72)発明者 ティアン, ギャング

アメリカ合衆国 4 4 1 4 5 オハイオ州 ウェストレイク ウィンドゲート コート 2 6 2 8

F ターム(参考) 5H004 GA04 GB20 HA07 HA10 HB08 JA04 LA12 LB06

5H303 BB01 BB06 CC01 DD01 EE07 JJ01

【外国語明細書】

LSTP102WO

## **METHOD FOR AUTOMATICALLY ESTIMATING INERTIA IN A MECHANICAL SYSTEM AND FOR GENERATING A MOTION PROFILE**

### **CROSS-REFERENCE TO RELATED APPLICATIONS**

**[0001]** This application claims the benefit of priority U.S. Application Serial No. 13/451,924 filed April 20, 2012 and U.S. Application Serial No. 13/474,919 filed May 18, 2012, the entire contents of both applications are incorporated herein by reference.

### **TECHNICAL FIELD**

**[0002]** This disclosure generally relates to motion control, and specifically to estimation of inertias and friction coefficients for use as parameters in a motion control system and generation of constraint-based, time-optimal motion profiles.

### **BACKGROUND**

**[0003]** Many automation applications employ motion control systems to control machine position and speed. Such motion control systems typically include one or more motors or similar actuating devices operating under the guidance of a controller, which sends position and speed control instructions to the motor in accordance with a user-defined control algorithm or program. Some motion control systems operate in a closed-loop configuration, whereby the controller instructs the motor to move to a target position or to transition to a target velocity (a desired state) and receives feedback information indicating an actual state of the motor. The controller monitors the feedback information to determine whether the motor has reached the target position or velocity, and adjusts the control signal to correct errors between the actual state and the desired state.

**[0004]** Designers of motion control systems seek to achieve an optimal trade-off between motion speed and system stability. For example, if the controller commands the motor to transition a mechanical component to a target position at a high torque, the machine may initially close the distance between the current position and the desired position at high speed (and thus

## LSTP102WO

in a time-efficient manner), but is likely to overshoot the desired position because of the high torque. Consequently, the controller must apply a corrective signal to bring the machine back to the desired position. It may take several such iterations before the motion system converges on the desired position, resulting in undesired machine oscillations. Conversely, instructing the motor to move at a lower torque may increase the accuracy of the initial state transition and reduce or eliminate machine oscillation, but will increase the amount of time required to place the machine in the desired position. Ideally, the controller gain coefficients should be selected to optimize the trade-off between speed of the state transition and system stability. The process of selecting suitable gain coefficients for the controller is known as tuning.

**[0005]** The response of a controlled mechanical system to a signal from a controller having a given set of controller gain coefficients depends on physical characteristics of the mechanical system, including the inertia and the friction coefficient. The inertia represents the resistance of the motion system to acceleration or deceleration. The friction coefficient represents a friction seen by the motor, such as the friction between the rotor and the shaft. Accurate estimates for the inertia and friction coefficient of a controlled mechanical system can simplify the tuning process and improve performance of the system. However, identifying accurate values for these parameters for a given mechanical system can be difficult. In some cases, the inertia is estimated using manual calculations based on the rated motor data and physical data (weight, dimensions, *etc.*) of the components comprising the load. Such calculations can be cumbersome and time consuming, and may not yield accurate values for these important parameters.

**[0006]** In another aspect of motion control, motion profiles are often used to facilitate transition between position or velocity states. For example, when the controller determines that the motion system must move to a new position or alter its velocity (*e.g.*, in accordance with the control algorithm or a user request), the controller must calculate a position or velocity trajectory – referred to as a motion profile – for transitioning the motion system from its current position/velocity to the target position/velocity. The motion profile defines the motion system's velocity, acceleration, and/or position over time

## LSTP102WO

as the system moves from the current state to the target state. Once this motion profile is calculated, the controller translates the motion profile into appropriate control signaling for moving the motion system through the trajectory defined by the profile.

**[0007]** In some applications, the various segments (or stages) of the motion profile are calculated based on predetermined user-defined constraints (e.g., maximum velocity, maximum acceleration, *etc.*), where the defined constraints may correspond to mechanical limitations of the motion system. Given these constraints and the desired target position and/or velocity, the controller will calculate the motion profile used to carry out the desired move or velocity change. The resultant motion profile is also a function of the type of profile the controller is configured to generate – typically either a trapezoidal profile or an S-curve profile. For a trapezoidal profile, the controller will calculate the motion profile according to three distinct stages— an acceleration stage, a constant velocity stage, and a deceleration stage. Such a profile results in a trapezoidal velocity curve. The S-curve profile type modifies the trapezoidal profile by adding four additional stages corresponding to these transitions. These additional stages allow gradual transitions between the constant (or zero) velocity stages and the constant acceleration/deceleration stages, providing smoother motion and affording a finer degree of control over the motion profile.

**[0008]** Since the trapezoidal profile always accelerates or decelerates at the maximum defined acceleration rate, this profile type tends to achieve faster point-to-point motion relative to S-curve profiles. However, since the transitions between the constant (or zero) velocity and the acceleration stages are abrupt, the trapezoidal curve may cause excessive system jerk at these transitions. Moreover, there is greater risk of overshooting the target position or velocity when using a trapezoidal motion profile, which can reduce accuracy or cause the controller to expend additional work and settling time bringing the motion device back to the desired target. Alternatively, the S-curve profile can yield greater accuracy due to the more gradual transitions between the constant velocity and acceleration/deceleration phases, but at the cost of additional time spent on the initial point-to-point move.

LSTP102WO

**[0009]** The above-described is merely intended to provide an overview of some of the challenges facing conventional motion control systems. Other challenges with conventional systems and contrasting benefits of the various non-limiting embodiments described herein may become further apparent upon review of the following description.

#### SUMMARY

**[0010]** The following presents a simplified summary of one or more embodiments in order to provide a basic understanding of such embodiments. This summary is not an extensive overview of all contemplated embodiments, and is intended to neither identify key or critical elements of all embodiments nor delineate the scope of any or all embodiments. Its purpose is to present some concepts of one or more embodiments in a simplified form as a prelude to the more detailed description that is presented later.

**[0011]** One or more embodiments of the present disclosure relate to systems and methods for automatically estimating inertia and friction coefficients for controlled mechanical systems. To this end, an inertia estimating system can instruct a controller to send a torque control signal to a motor, where the torque control signal varies continuously over time between defined maximum and minimum torque values. This torque control signal can be controlled based on a testing sequence defined in the inertia estimating system. In a non-limiting example, the testing sequence can specify that the torque control signal will increase gradually at a defined rate of increase, causing the motor to accelerate. In response to a defined trigger, the torque control signal can gradually decrease back to zero, causing the motor to decelerate to a rest state.

**[0012]** During these acceleration and deceleration phases, the inertia estimating system can measure and record the velocity of the motor over time in response to the torque control signal. An inertia component can then determine one or both of an estimated inertia and an estimated friction coefficient for the mechanical system based on the time-varying torque signal and the measured velocity curve. The estimated inertia and/or the friction

LSTP102WO

coefficient can subsequently be used by the controller to facilitate identification of appropriate controller gains for the system.

**[0013]** Other aspects of the present disclosure relate to systems and methods for efficiently generating constraint-based, time-optimal motion profiles. To this end, a profile generator deployed within a controller can leverage a mathematical algorithm to solve for constraint-based, time-optimal point-to-point motion in real-time and to calculate trajectories based on the solution. To achieve smooth and accurate point-to-point motion, the profile generator can calculate the trajectory based on an ST-curve profile type, which generates profiles having a continuous jerk reference over time for at least one acceleration or deceleration segment of the profile. By calculating motion profiles that includes a time-varying jerk reference, the profile generator of the present disclosure can yield smoother and more stable motion compared to traditional trapezoidal or S-curve profiles.

**[0014]** The ST-curve profiles generated by the profile generator can support asymmetric acceleration and deceleration phases. Conventionally, asymmetric acceleration and deceleration is supported only by trapezoidal profiles, but not by the smoother S-curve profiles. The ST-curves generated according to the techniques described herein can allow asymmetric acceleration and deceleration to be used with smoother motion profiles. In some embodiments, the profile generator described herein may also generate S-curve profiles that support asymmetric acceleration and deceleration.

**[0015]** In another aspect, one or more embodiments of the profile generator described herein can improve calculation efficiency by omitting calculations for trajectory segments that will not be used in the final trajectory. That is, rather than calculating profile data for all seven profile stages even in cases for which one or more of the segments will not be used, the profile generator described herein may calculate only those profile stages that will be used in the final motion profile for a given trajectory, reducing computational overhead within the controller. The profile generator can automatically determine which segment(s) of the motion profile may be skipped for a given point-to-point move and calculate the remaining segments accordingly.

**[0016]** According to another aspect, one or more embodiments of the profile generator described herein can further improve the accuracy and

LSTP102WO

efficiency of a point-to-point move by forcing the total profile time to be a multiple of the motion controller's sample time. In an exemplary technique, the profile generator can calculate a time-optimal solution for a given point-to-point move, determine the time durations of the respective segments of the resultant profile, and round these durations to be multiples of the sample time. The profile generator can then recalculate the jerk, acceleration/deceleration, velocity and/or position references for the profile to be consistent with these rounded profile times. Thus, the trajectory outputs can be aligned with the sample points, mitigating the need to compensate for small differences introduced when the total profile time falls between two sample times.

**[0017]** The following description and the annexed drawings set forth herein detail certain illustrative aspects of the one or more embodiments. These aspects are indicative, however, of but a few of the various ways in which the principles of various embodiments can be employed, and the described embodiments are intended to include all such aspects and their equivalents.

#### BRIEF DESCRIPTION OF THE DRAWINGS

**[0018]** FIG. 1 is a block diagram of a simplified closed-loop motion control architecture.

**[0019]** FIG. 2 is a block diagram of an exemplary non-limiting inertia estimating system.

**[0020]** FIG. 3 is a block diagram illustrating the inputs and outputs associated with the inertia estimator.

**[0021]** FIG. 4 is a block diagram depicting the interactions between the inertia estimator and a motion control system during an exemplary test sequence.

**[0022]** FIG. 5 illustrates an exemplary torque command  $u(t)$  and corresponding velocity feedback  $v(t)$  graphed over time.

**[0023]** FIG. 6 is a block diagram depicting an inertia estimator having an inertia component and friction coefficient component.

**[0024]** FIG. 7 is a block diagram of an exemplary configuration in which an inertia estimator operates as an independent component relative to a motion controller.



LSTP102WO

**[0025]** FIG. 8 illustrates an exemplary motion control tuning application that utilizes the estimated inertia and friction coefficient generated by the inertia estimator.

**[0026]** FIG. 9 is a flowchart of an example methodology for estimating an inertia and a friction coefficient for a controlled mechanical system.

**[0027]** FIG. 10 is a flowchart of an example methodology for executing a testing sequence on a motion control system in order to estimate the inertia and friction coefficient.

**[0028]** FIG. 11 is a block diagram of an exemplary motion profile generating system capable of generating motion profiles in a motion control system.

**[0029]** FIG. 12 is a block diagram of exemplary motion controller that utilizes a profile generator.

**[0030]** FIG. 13 is a block diagram illustrating the inputs and outputs of an exemplary position profile generator.

**[0031]** FIG. 14 is a block diagram illustrating the inputs and outputs of an exemplary velocity profile generator.

**[0032]** FIG. 15 is a graphical comparison of an exemplary ST-curve profile with traditional trapezoidal and S-curve profiles.

**[0033]** FIG. 16 illustrates an exemplary S-curve profile that utilizes all seven stages.

**[0034]** FIG. 17 illustrates an exemplary S-curve motion profile that skips the constant velocity stage.

**[0035]** FIG. 18 illustrates an example S-curve profile that skips the constant acceleration and constant deceleration stages.

**[0036]** FIG. 19 illustrates an example S-curve profile that skips the constant acceleration, constant velocity, and constant deceleration stages.

**[0037]** FIG. 20 is a flowchart of an example methodology for calculating a motion profile for a point-to-point move in a motion control system.

**[0038]** FIG. 21 is a flowchart of an example methodology for calculating a constraint-based time-optimal motion profile that conforms to a sample time of a motion controller.

**[0039]** FIG. 22 is a flowchart of an example methodology for calculating a motion profile for a point-to-point move using segment skipping.



LSTP102WO

**[0040]** FIG. 23 is a block diagram representing an exemplary networked or distributed computing environment for implementing one or more embodiments described herein.

**[0041]** FIG. 24 is a block diagram representing an exemplary computing system or operating environment for implementing one or more embodiments described herein.

#### DETAILED DESCRIPTION

**[0042]** Various embodiments are now described with reference to the drawings, wherein like reference numerals refer to like elements throughout. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide an understanding of this disclosure. It is to be understood, however, that such embodiments may be practiced without these specific details, or with other methods, components, materials, *etc.* In other instances, structures and devices are shown in block diagram form to facilitate describing one or more embodiments.

**[0043]** Systems and methods described herein relate to techniques for generating estimated inertia and friction coefficients for controlled mechanical systems. One or more embodiments of the present disclosure can estimate these parameters in a substantially automated fashion by running the mechanical system through a testing sequence to be defined in more detail herein. Results of this testing sequence can be used to generate accurate inertia and friction coefficient estimates for the system. These estimated parameters can subsequently be used to facilitate simplified and accurate tuning and control of the motion system.

#### **[0044]** Inertia and Friction Coefficient Estimation

**[0045]** FIG. 1 depicts a simplified closed-loop motion control architecture. Controller 102 is programmed to control motor 104, which drives mechanical load 106. Controller 102, motor 104, and load 106 comprise the primary components of an exemplary motion control system. In an exemplary non-limiting application, load 106 can represent an axis of a single- or multi-axis robot or positioning system. In such applications, controller 102 sends

## LSTP102WO

control signal 108 instructing the motor 104 to move the load 106 to a desired position at a desired speed. The control signal 108 can be provided directly to the motor 104, or to a motor drive (not shown) that controls the power delivered to the motor 104 (and consequently the speed and direction of the motor). Feedback signal 110 indicates a current state (*e.g.*, position, velocity, *etc.*) of the motor 104 and/or load 106 in substantially real-time. In servo-driven systems, feedback signal 110 can be generated, for example, by an encoder or resolver (not shown) that tracks an absolute or relative position of the motor. In sensorless systems lacking a velocity sensor, the feedback signal can be provided by a speed/position estimator. During a move operation, the controller monitors feedback signal 110 to ensure that the load 106 has accurately reached the target position. The controller 102 compares the actual position of the load as indicated by the feedback signal 110 with the target position, and adjusts the control signal 108 as needed to reduce or eliminate error between the actual and target positions.

**[0046]** In another exemplary application, load 106 can represent a spinning load (*e.g.*, a pump, a washing machine, a centrifuge, *etc.*) driven by motor 104, in which controller 102 controls the rotational velocity of the load. In this example, controller 102 provides an instruction to motor 104 (via control signal 108) to transition from a first velocity to a second velocity, and makes necessary adjustments to the control signal 108 based on feedback signal 110. It is to be appreciated that the parameter estimation techniques of the present application are not limited to use with the exemplary types of motion control systems described above, but are applicable for any suitable motion control application.

**[0047]** The control signal output generated by the controller 102 in response to an error between the desired position or velocity and the target position or velocity (as reported by the feedback signal 110) depends on the gain coefficients for the control loop. Design engineers must often employ a trial-and-error approach to identifying suitable gain coefficients (*i.e.* tuning the control loop), since suitable gain selection depends on physical characteristics of the mechanical system being controlled. For example, mechanical systems with a high inertia (resistance to acceleration or deceleration) may require relatively high initial torque to initiate a move to a new position or velocity,

LSTP102WO

particularly if the application requires rapid convergence on the target position/velocity. However, high torque commands increase the possibility of overshoot, necessitating a reverse correction to bring the system back to the target. Non-optimal gain settings can result in undesired mechanical oscillations as the system performs multiple corrective iterations before settling on the target position or velocity. Such oscillations can introduce instability, cause system delays, and consume excessive power as a result of the additional work required to bring the system to a stable state. The friction of the motor can also affect how the mechanical system responds to a given control signal, and is therefore a factor to be considered when tuning the control system.

**[0048]** Control system tuning can be simplified if accurate estimates of the mechanical system's inertia and friction coefficient are known. Knowledge of these parameters can also improve performance of the system during operation. Accordingly, one or more embodiments of the present application can accurately estimate a controlled mechanical system's inertia and friction coefficient in a substantially automated fashion.

**[0049]** FIG. 2 is a block diagram of an exemplary non-limiting inertia estimating system capable of generating estimated values of a mechanical system's inertia and friction coefficient. Inertia estimator 202 can include a torque command generator 204, a velocity monitoring component 206, an inertia component 208, a friction coefficient component 210, an interface component 212, one or more processors 214, and memory 216. In various embodiments, one or more of the torque command generator 204, velocity monitoring component 206, inertia component 208, friction coefficient component 210, interface component 212, the one or more processors 214, and memory 218 can be electrically and/or communicatively coupled to one another to perform one or more of the functions of the inertia estimator 202. In some embodiments, components 204, 206, 208, 210, and 212 can comprise software instructions stored on memory 216 and executed by processor(s) 214. The inertia estimator 202 may also interact with other hardware and/or software components not depicted in FIG. 2. For example, processor(s) 214 may interact with one or more external user interface device,

LSTP102WO

such as a keyboard, a mouse, a display monitor, a touchscreen, or other such interface devices.

**[0050]** Interface component 212 can be configured to receive user input and to render output to the user in any suitable format (e.g., visual, audio, tactile, *etc.*). User input can be, for example, user-entered parameters used by the inertia estimator when executing an inertia estimation sequence (to be described in more detail below). Torque command generator 204 can be configured to output a torque control command that varies continuously over time according to a defined testing sequence. Velocity monitoring component 206 can receive velocity data for the mechanical system for use in calculating the inertia and friction coefficient. In some embodiments, the velocity monitoring component 206 can measure and record the velocity of the motor over time in response to the applied torque control command generated by the torque command generator 204. Alternatively, the velocity monitoring component 206 can receive the measured velocity data from separate measuring instrumentation. Inertia component 208 and friction coefficient component 210 can be configured to calculate an inertia and a friction coefficient, respectively, based on the time-varying torque command generated by torque command generator 204 and the measured velocity curve acquired by the velocity monitoring component 206. The one or more processors 214 can perform one or more of the functions described herein with reference to the systems and/or methods disclosed. Memory 216 can be a computer-readable storage medium storing computer-executable instructions and/or information for performing the functions described herein with reference to the systems and/or methods disclosed.

**[0051]** The inertia estimator can generate estimates for a mechanical system's inertia and friction coefficient by running the system through a testing sequence and calculating the estimates based on the results. FIG. 3 is a block diagram illustrating the inputs and outputs associated with inertia estimator 302 (similar to inertia estimator 202). Inertia estimator 302 can generate a torque command 310, which instructs a motor driving the motion system to rotate in a specified direction at a given torque. Rather than issuing one or more constant torque commands that transition between constant torque values in sudden steps (resulting in a step-shaped torque output),

LSTP102WO

inertia estimator 302 can control torque command 310 such that the torque value varies continuously over time between a maximum and minimum torque value. Inertia estimator 302 controls the torque value issued via torque command 310 in accordance with a testing sequence having user-defined parameters, as will be discussed in more detail below.

**[0052]** The motion system will accelerate or decelerate in accordance with the torque command 310 issued by inertia estimator 302, and velocity feedback 304 from the motion control system is provided to inertia estimator 302. Velocity feedback 304 represents the velocity of the motion system over time in response to application of torque command 310. In an exemplary testing sequence, inertia estimator 302 can control torque command 310 as a function of the velocity feedback 304 and one or more user-defined setpoints. The user-defined setpoints can include one or more torque limits 306 defining the upper and lower bounds of the torque command signal, and/or one or more velocity checkpoints 308 defining trigger velocity values used to control the torque command 310 and generate the estimates.

**[0053]** Upon completion of the testing sequence, inertia estimator 302 generates an estimate of the motion system's inertia 312 and/or an estimate of the motion system's friction coefficient 314. Inertia estimator 302 determines these estimates based on the torque command 310 that was issued to the motion system and the corresponding velocity feedback 304. In one or more embodiments, inertia estimator 302 can integrate selected portions of the torque curve (corresponding to torque command 310) and the velocity curve (corresponding to velocity feedback 304) over time, and calculate the inertia estimate 312 and the friction coefficient estimate 314 as functions of these integrals.

**[0054]** FIG. 4 illustrates the interactions between inertia estimator and a motion control system during an exemplary testing sequence. In this example, motion system 424 comprises a motor 424, which responds to control signaling 420 provided by controller 418. Motor 424 is used to drive a load (not shown), such as a positioning axis, a rotational component of a machine, or other motor-driven load. Controller 418 also monitors feedback 422, which provides substantially real-time state data for the motor 424 (*e.g.*, position, speed, *etc.*).



LSTP102WO

**[0055]** In the illustrated example, inertia estimator 402 is depicted as a separate element from controller 418 for clarity. For such configurations, inertia estimator 402 can exchange data with controller 418 or other elements of the motion system 424 via any suitable communications means, including but not limited to wired or wireless networking, hardwired data links, or other such communication means. In other embodiments, inertia estimator 402 can be an integrated component of controller 418. For example, inertia estimator 402 can be a functional component of the controller's operating system and/or control software executed by one or more processors residing on the controller 418. Inertia estimator 402 can also be a hardware component residing within controller 418, such as a circuit board or integrated circuit, that exchanges data with other functional elements of the controller 418. Other suitable implementations of inertia estimator 402 are within the scope of certain embodiments of the present disclosure.

**[0056]** Prior to testing, one or more user-defined parameters 412 are provided to inertia estimator 402 via interface component 406 (similar to interface component 212 described in connection with FIG. 2). These parameters can include a maximum torque  $u_{max}$  and a minimum torque  $u_{min}$  defining upper and lower limits on the torque command to be generated by torque command generator 408 (similar to torque command generator 204 of FIG. 2). In some embodiments, the inertia estimator 402 may only require the maximum torque  $u_{max}$  to be defined by the user, and can use the magnitude of the defined maximum torque as a limiting value for both the forward and reverse directions. In other embodiments, inertia estimator 402 may accept values for both  $u_{max}$  and  $u_{min}$ , allowing for different torque setpoints for the forward and reverse directions, respectively. The values selected for  $u_{max}$  and  $u_{min}$  can correspond to the expected operational limits of the motion system 424, thereby allowing the inertia and friction coefficient to be determined based on characteristics of the motion system 424 over the system's entire torque profile. User-defined parameters 412 can also include one or more velocity checkpoints ( $v_1$ ,  $v_2$ ,  $v_3$ ...) defining critical velocities used to define stages of the test sequence, as will be described in more detail below.

LSTP102WO

**[0057]** Interface component provides torque command generator 408 with the user-defined parameters 412. When testing is initiated, torque command generator 408 outputs a torque command 414 to the motion system 424. Torque command 414 is represented as  $u(t)$ , since the torque command generator 408 will vary the torque command continuously over time. In the configuration depicted in FIG. 4, inertia estimator 402 sends torque command 414 to controller 418, which in turn instructs the motor 424 (via control signaling 420) to rotate in the indicated direction at the indicated torque. As the motor is rotating, velocity monitoring component 410 reads velocity data 416 from controller 418 (which itself measures the velocity of the motor 424 via feedback 422). The measured velocity 416 over time is represented as  $v(t)$ .

**[0058]** As testing proceeds, torque command generator 408 can vary the torque command 414 in accordance with a predefined testing sequence, wherein phases of the testing sequence are triggered by the velocity feedback 416 relative to the user-defined parameters 412. An exemplary testing sequence is now explained with reference to FIG. 5, which illustrates an exemplary torque command  $u(t)$  and corresponding velocity feedback  $v(t)$  graphed over time. As shown on torque graph 502, the torque command signal  $u(t)$  is bounded by  $u_{max}$  and  $u_{min}$ . Velocity checkpoints  $v1$ ,  $v2$ , and  $v3$ , shown on velocity graph 504, will determine phase transitions of the testing sequence. The values of  $u_{max}$ ,  $u_{min}$ ,  $v1$ ,  $v2$ , and  $v3$  can be defined by the user prior to testing (e.g., as user-defined parameters 412 of FIG. 4).

**[0059]** When testing begins at time  $t = 0$ , the applied torque signal  $u(t)$  and the motor velocity  $v(t)$  are both zero. Initially, torque command generator sends a negative torque signal to the motion system, causing the motion system to accelerate in the negative direction. For the first phase of this exemplary test, the torque command generator gradually decreases the torque command  $u(t)$  until the velocity of the motor  $v(t)$  reaches  $v1$  or until the torque command  $u(t)$  reaches  $u_{min}$ . In the present example, the motor velocity  $v(t)$  reaches  $v1$  at time  $t = t1$ , triggering the second phase of the test. As shown on graph 502, the torque command  $u(t)$  is decreased continuously at a substantially constant rate between time  $t = 0$  and  $t = t1$ . In one or more

LSTP102WO

embodiments, the rate at which torque command is decreased or increased (that is, the slope of  $u(t)$ ) can be configured as a user-defined parameter of the inertia estimator 402 (e.g., via interface component 406).

**[0060]** For the second phase of the test (starting at time  $t = t_1$ ), the torque command generator gradually increases the torque command  $u(t)$  until either the motor velocity  $v(t)$  reaches velocity checkpoint  $v_3$  or the torque command  $u(t)$  reaches the torque setpoint  $u_{max}$ . In the present example, the torque command  $u(t)$  reaches the upper limit  $u_{max}$  before the motor velocity  $v(t)$  reaches velocity checkpoint  $v_3$ . Since the motor is still accelerating at this time, the torque command generator maintains the torque command signal at  $u_{max}$  until the velocity  $v(t)$  reaches  $v_3$ . As illustrated on velocity graph 504, the motor velocity reaches  $v_3$  at time  $t = t_4$ . If the velocity  $v(t)$  does not reach velocity checkpoint  $v_3$  within a defined timeout period after torque command signal has reached  $u_{max}$  (e.g., if velocity checkpoint  $v_3$  was inadvertently set higher than the physical velocity limit of the motion system), the inertia estimator can initiate a suitable timeout handling routine. This timeout handling routine can comprise, for example, aborting the testing sequence and displaying an error message via interface component 406.

**[0061]** As the motion system is accelerating toward  $v_3$  during this phase, the velocity passes through velocity checkpoint  $v_2$ , which denotes the acceleration phase of the testing sequence. Velocity checkpoint  $v_2$  is set to be greater than zero and less than  $v_3$ , and is used to delineate the beginning of the acceleration phase of the testing sequence and the end of the deceleration phase, as will be discussed in more detail below.

**[0062]** Upon determining that the motor velocity has reached  $v_3$ , the torque command generator begins the third phase of the test at time  $t = t_4$  by gradually decreasing the torque command  $u(t)$ . As the torque command  $u(t)$  is decreased, the motor will continue to accelerate for a brief time until the value of the torque command  $u(t)$  becomes less than the friction force of the motion system, at which time the motor will begin to decelerate. Since the motor was still accelerating when the velocity reached  $v_3$  at time  $t = t_3$ , the velocity will continue past  $v_3$  for some time after the torque command begins decreasing. In accordance with the testing sequence definition, the torque



LSTP102WO

command generator continues to decrease  $u(t)$  until the motor velocity  $v(t)$  returns to velocity checkpoint  $v_3$  (at time  $t = t_6$ ), and thereafter holds  $u(t)$  constant until the motor velocity  $v(t)$  returns to velocity checkpoint  $v_2$  (at time  $t = t_7$ ). At this point, the inertia estimator has the data it requires to calculate estimates for the inertia and friction coefficient for the mechanical system. The torque command generator therefore brings the torque command signal  $u(t)$  back to zero (at  $t = t_8$ ), allowing the motion system to coast to a resting state, as illustrated on graph 504 by the tapering end of the  $v(t)$  curve.

**[0063]** The testing sequence described above in connection with FIG. 5 is only intended to represent an exemplary, non-limiting testing sequence. It is to be understood that any suitable testing sequence that continuously varies the torque command  $u(t)$  over time and measures a corresponding velocity profile  $v(t)$  for the motion system is within the scope of certain embodiments of this disclosure. For example, although the foregoing example describes the torque command  $u(t)$  as changing direction in response to the velocity  $v(t)$  reaching the respective velocity checkpoints, some test sequences may include phases in which the torque command  $u(t)$  only changes its rate of increase or decrease when the velocity checkpoint is reached, without altering the direction of the torque command (e.g., an increasing torque command may continue to increase in response to  $v(t)$  reaching a phase checkpoint, but at a slower rate).

**[0064]** As the foregoing testing sequence is performed, the inertia estimator 402 records both the torque command signal  $u(t)$  generated by torque command generator 408 and the corresponding motor velocity  $v(t)$  read by the velocity monitoring component 410. These torque and velocity curves characterize the motion system 424 such that accurate estimates of the inertia and friction coefficient can be calculated based on the curves. In one or more embodiments, inertia estimator calculates these estimates based on integrals of  $u(t)$  and  $v(t)$ . The following illustrates an exemplary, non-limiting technique for leveraging integrals of  $u(t)$  and  $v(t)$  to derive estimates for the inertia and friction coefficient for a motion system.

**[0065]** A motion system can be described by the differential equation:

LSTP102WO

**[0066]** 
$$J\dot{v}(t) = -Bv(t) + u(t) \quad (1)$$

**[0067]** where  $J$  is the inertia,  $B$  is the friction coefficient,  $u(t)$  is the torque command signal, and  $v(t)$  is the corresponding velocity of the motion system in response to the torque signal  $u(t)$  (e.g.,  $u(t)$  and  $v(t)$  described above in connection with FIGs. 4 and 5).

**[0068]** Integrating both sides of equation (1) for respective acceleration and deceleration stages yields:

**[0069]** 
$$J\Delta v_{acc}(t) = -B \int v_{acc}(t) + \int u_{acc}(t) \quad (2)$$

**[0070]** 
$$J\Delta v_{dec}(t) = -B \int v_{dec}(t) + \int u_{dec}(t) \quad (3)$$

**[0071]** where  $u_{acc}(t)$  and  $v_{acc}(t)$  are portions of  $u(t)$  and  $v(t)$ , respectively, corresponding to the acceleration phase of the testing sequence, and  $u_{dec}(t)$  and  $v_{dec}(t)$  are portions of  $u(t)$  and  $v(t)$ , respectively, corresponding to the deceleration phase.

**[0072]** Equations (2) and (3) can be solved to yield estimates of the inertia  $J$  and friction coefficient  $B$ :

**[0073]** 
$$J = \frac{\int u_{dec}(t) \int v_{acc}(t) - \int u_{acc}(t) \int v_{dec}(t)}{\Delta v_{dec}(t) \int v_{acc}(t) - \Delta v_{acc}(t) \int v_{dec}(t)} \quad (4)$$

**[0074]** 
$$B = \frac{\Delta v_{dec}(t) \int u_{acc}(t) - \Delta v_{acc}(t) \int u_{dec}(t)}{\Delta v_{dec}(t) \int v_{acc}(t) - \Delta v_{acc}(t) \int v_{dec}(t)} \quad (5)$$

**[0075]** For the exemplary torque and velocity curves depicted in FIG. 4, the acceleration phase is taken to be the period starting when velocity  $v(t)$  reaches velocity checkpoint  $v_2$  for the first time (at time  $t = t_3$ ) and ending when torque signal  $u(t)$  crosses zero (at time  $t = t_5$ ). The velocity of the motion system at the end of this acceleration phase is recorded as  $v_4$  (as indicated on graph 504). The deceleration phase is taken to be the period starting when torque signal  $u(t)$  crosses zero (at time  $t = t_5$ ) and ending when

LSTP102WO

velocity  $v(t)$  returns to  $v_2$  (at time  $t = t_7$ ). The inertia estimator 402 can be configured to recognize these acceleration and deceleration phase delineations in order to derive the estimated inertia and friction coefficient based on equations (4) and (5) above. It is to be appreciated that other criteria for delineating the acceleration and deceleration phases are also within the scope at certain embodiments of this disclosure.

**[0076]** Given these acceleration and deceleration phase definitions, the integrals of  $u_{acc}(t)$  and  $u_{dec}(t)$  are represented by the shaded regions of graph 502 labeled  $U_{acc}$  and  $U_{dec}$ , respectively, and the integrals of  $v_{acc}(t)$  and  $v_{dec}(t)$  are represented as the shaded regions of graph 504 labeled  $V_{acc}$  and  $V_{dec}$ , respectively. Accordingly,  $U_{acc}$ ,  $U_{dec}$ ,  $V_{acc}$ , and  $V_{dec}$  are defined as follows:

$$\textbf{[0077]} \quad U_{acc} = \int u_{acc}(t) \quad (6)$$

$$\textbf{[0078]} \quad U_{dec} = \int u_{dec}(t) \quad (7)$$

$$\textbf{[0079]} \quad V_{acc} = \int v_{acc}(t) \quad (8)$$

$$\textbf{[0080]} \quad V_{dec} = \int v_{dec}(t) \quad (9)$$

**[0081]** Substituting equations (6) – (9) into equations (4) and (5), the inertia  $J$  and friction coefficient  $B$  can be represented as:

$$\textbf{[0082]} \quad J = \frac{U_{dec}V_{acc} - U_{acc}V_{dec}}{\Delta v_{dec}(t)V_{acc} - \Delta v_{acc}(t)V_{dec}} \quad (12)$$

$$\textbf{[0083]} \quad B = \frac{\Delta v_{dec}(t)U_{acc} - \Delta v_{acc}(t)U_{dec}}{\Delta v_{dec}(t)V_{acc} - \Delta v_{acc}(t)V_{dec}} \quad (13)$$

**[0084]** where the velocity deltas  $\Delta v_{acc}(t)$  and  $\Delta v_{dec}(t)$  are defined as:

$$\textbf{[0085]} \quad \Delta v_{acc}(t) = v_4 - v_2 \quad (10)$$

LSTP102WO

**[0086]**  $\Delta v_{dec}(t) = v_2 - v_4$  (11)

**[0087]** Equations (12) and (13) are exemplary, non-limiting formulas for calculating an estimated inertia and friction coefficient for a motion system based on continuous torque and velocity data. It is to be appreciated that any suitable formula for calculating these parameters through integration of a continuous torque signal and a corresponding velocity curve are within the scope of certain embodiments of this disclosure.

**[0088]** Upon completion of the testing sequence described above in connection with FIGs. 4 and 5, the inertia estimator can apply equations (12) and (13) (or other suitable formulas) to the continuous torque data  $u(t)$  and motor velocity data  $v(t)$  acquired by the test to derive estimates for the inertia and friction coefficient. FIG. 6 is a block diagram depicting an inertia estimator 602 having an inertia component 606 and friction coefficient component 608 according to one or more embodiments of the present disclosure. Although inertia estimator 602 is depicted as including both an inertia component 606 and a friction coefficient component 608, it is to be appreciated that some embodiments of the inertia estimator 602 may include only one of these components without deviating from the scope of the present disclosure. That is, the inertia estimator 602 may be configured to calculate one or both of the inertia or the friction coefficient.

**[0089]** After the torque data  $u(t)$  and velocity data  $v(t)$  have been obtained, the torque command generator 604 (similar to torque command generator 408 and 204) provides the torque data to inertia component 606 and friction coefficient component 608 (similar to inertia component 208 and friction coefficient component 210, respectively, of FIG. 2). Similarly, the velocity monitoring component 606 can provide the acquired velocity data  $v(t)$  to inertia component 606 and friction coefficient component 608. According to one or more embodiments, inertia estimator 602 can segregate the torque and velocity data into acceleration phase data ( $u_{acc}(t)$  and  $v_{acc}(t)$ ) and deceleration phase data ( $u_{dec}(t)$  and  $v_{dec}(t)$ ) so that values can be derived for  $U_{acc}$ ,  $U_{dec}$ ,  $V_{acc}$ , and  $V_{dec}$  according to equations (6)-(9) above.

**[0090]** Inertia component 606 can integrate  $u_{acc}(t)$ ,  $u_{dec}(t)$ ,  $v_{acc}(t)$ , and  $v_{dec}(t)$  and calculate the estimated inertia  $J$  610 as a function of the

## LSTP102WO

integrals (e.g., based on equation (12) or variation thereof). Similarly, friction coefficient component 608 can calculate the estimated friction coefficient  $B$  616 as a function of the integrals (e.g., based on equation (13)). Inertia estimator 602 can then output the estimated inertia  $J$  610 and friction coefficient  $B$  612 according to the requirements of a particular application in which the inertia estimator operates. For example, inertia estimator 602 may provide inertia  $J$  610 and friction coefficient  $B$  612 to a motion controller 614, which can use the values of  $J$  and  $B$  to facilitate tuning one or more gain coefficients. Inertia estimator 602 may also output the estimated values for  $J$  and  $B$  to a display (e.g., via interface component 212) so that the values can be viewed and entered manually into a separate motion control or tuning application. Accurate estimates of the motion system's inertia  $J$  610 and friction coefficient  $B$  612 can simplify the tuning process and facilitate accurate parameter tuning, resulting in precise and energy-efficient machine motion. Moreover, since the inertia estimator calculates values for  $J$  and  $B$  based on data collected over the motion system's entire torque profile (rather than extrapolating based on the system's response to one or more constant torque commands), the inertia and friction coefficient estimates derived by the inertia estimator are more likely to be accurate over the full operational range of the motion system.

**[0091]** While the preceding examples have described the inertia estimator as sending the torque command  $u(t)$  and receiving velocity feedback  $v(t)$  via the motion controller (e.g., controller 418 of FIG. 4), either as a separate component operating through the controller or as an integrated component of the controller, other configurations are within the scope of certain embodiments of this disclosure. For example, FIG. 7 illustrates an architecture in which inertia estimator 706 operates as an independent separate component from controller 702. In this exemplary architecture, inertia estimator 706 is capable of generating its own torque command signal independently of controller 702. The motor 704 being tested and controlled can receive its torque command signal 708 from either controller 702 or inertia estimator 706 depending on the state of switch 712. The velocity feedback 710 from the motor 704 can be provided to both the controller 702 and inertia estimator 706. During the testing sequence, switch 712 can be set to convey

## LSTP102WO

the torque command  $u(t)$  from inertia estimator 706. Testing can proceed as described in previous examples, such that inertia estimator 706 generates estimated values for the inertia  $J$  and friction coefficient  $B$  for the motion system. Inertia estimator 706 can then provide the estimated values for  $J$  and  $B$  to the controller 702, which can use these values to determine suitable controller gain coefficients or other control parameters. Once the controller parameters have been set, switch 712 can be positioned to provide torque command 708 from controller 702 to the motor 704, and normal operation of the motion system can be carried out using the controller gain coefficients derived based on  $J$  and  $B$ .

**[0092]** FIG. 8 illustrates an exemplary motion control tuning application that utilizes the estimated inertia and friction coefficient generated by the inertia estimator. In this example, a tuning application 804 is used to tune the controller gains for controller 806, where the controller 806 controls operation of a motor-driven motion system (not shown). Inertia estimator 802 can generate estimates of the motion system's inertia  $J$  808 and friction coefficient  $B$  810 according to the techniques described above. Specifically, inertia estimator 802 instructs controller 806 to send a continuous torque command to the motion system's motor, where the torque command varies continuously over time according to a predefined testing sequence. Alternatively, for embodiments in which inertia estimator 802 operates independently of controller 806 (as in the exemplary configuration depicted in FIG. 7), the inertia estimator 802 can generate and send its own continuous torque command to the motion system. The testing sequence can include acceleration and deceleration phases, during which the inertia estimator 802 monitors and records the velocity of the motion system in response to the applied torque command. At the conclusion of the testing sequence, inertia estimator 802 can calculate estimates of inertia  $J$  808 and friction coefficient  $B$  810 based on integrals of the time-varying torque command signal and the corresponding time-varying motion system velocity (e.g., based on equations (12) and (13)).

**[0093]** Inertia estimator 802 can then provide inertia  $J$  808 and friction coefficient  $B$  810 to the tuning application 804. Alternatively, inertia estimator 802 can render the values of  $J$  and  $B$  on a user interface, allowing a user to



## LSTP102WO

manually enter the estimated inertia and friction coefficients into the tuning application 804. Knowledge of  $J$  and/or  $B$  can allow the tuning application 804 to generate suitable estimates for one or more controller gains 812 based on the mechanical properties of the motion system. Tuning application 804 can generate suitable values for controller gains 812 as a function of the inertia  $J$  and/or friction coefficient  $B$  810, as well as control system bandwidth (e.g., crossover frequency) 814, which can be manually adjusted by the user via interface 816 to achieve desired motion characteristics.

**[0094]** In typical applications, the inertia estimator described herein can be used to generate reliable estimates of a motion system's inertia  $J$  and friction coefficient  $B$  during initial deployment of the motion control system, prior to normal operation. Specifically, the inertia estimator can be used in connection with configuring and tuning the controller parameters (e.g., controller gain coefficients) prior to runtime. Once set, these parameters typically remain fixed after system startup, unless it is decided to re-tune the system at a later time. However, in some embodiments, the inertia estimator can be configured to automatically recalculate values for  $J$  and  $B$  periodically or continuously during runtime. Using such configurations, controller parameters that are based on estimates of  $J$  and  $B$  can be dynamically adjusted during normal operation, substantially in real-time, to compensate for gradual changes to the motion system's mechanical properties (e.g., as a result of mechanical wear and tear, changes to the load seen by a motor, etc.).

**[0095]** FIGS. 9-10 illustrate various methodologies in accordance with certain disclosed aspects. While, for purposes of simplicity of explanation, the methodologies are shown and described as a series of acts, it is to be understood and appreciated that the disclosed aspects are not limited by the order of acts, as some acts may occur in different orders and/or concurrently with other acts from that shown and described herein. For example, those skilled in the art will understand and appreciate that a methodology can alternatively be represented as a series of interrelated states or events, such as in a state diagram. Moreover, not all illustrated acts may be required to implement a methodology in accordance with certain disclosed aspects. Additionally, it is to be further appreciated that the methodologies disclosed

LSTP102WO

hereinafter and throughout this disclosure are capable of being stored on an article of manufacture to facilitate transporting and transferring such methodologies to computers.

**[0096]** FIG. 9 illustrates an example methodology 900 for estimating an inertia and a friction coefficient for a controlled mechanical system. At 902, a continuous torque command  $u(t)$  is sent to a controller of a motion system, where torque command  $u(t)$  varies over time between defined maximum and minimum torque setpoints. In one or more embodiments, the torque command  $u(t)$  can accord to a predefined testing sequence, such that the output of  $u(t)$  depends on the phase of the testing sequence and the response of the mechanical system relative to one or more user-defined setpoints. The test sequence can comprise both acceleration and deceleration phases, corresponding to increasing and decreasing motor speeds, respectively. For two-directional testing, the torque command  $u(t)$  can vary between positive and negative torque values during the testing sequence, causing the motion system to accelerate in both directions during the test.

**[0097]** At 904, the velocity  $v(t)$  of the motion system in response to the torque command  $u(t)$  is recorded. Thus, upon completion of the testing sequence, data curves for both the applied torque command  $u(t)$  and the resultant motion system velocity  $v(t)$  are obtained for  $t = 0 - t_{end}$ , where  $t_{end}$  is the duration of the test sequence.

**[0098]** At 906, estimates for at least one of the inertia or the friction coefficient of the motion system are calculated based on integrals of the torque curve  $u(t)$  and the velocity curve  $v(t)$ . In one or more embodiments, the curves for  $u(t)$  and  $v(t)$  can be divided into an acceleration phase and a deceleration phase, and the inertia and the friction coefficient can be calculated based on respective integrals of the acceleration and deceleration phases (e.g., using equations (12) and (13) above, or other suitable equation). At 908, one or more parameters for the motion system are set as a function of the estimated inertia and/or friction coefficient calculated at step 906. In a non-limiting example, one or more controller gain coefficients can be set



## LSTP102WO

based on the estimated inertia and/or friction coefficient calculated according to steps 902-906.

**[0099]** FIG. 10 illustrates an example methodology 1000 for executing a testing sequence on a motion control system in order to estimate the inertia and friction coefficient. At 1002, a torque command to the motion system is continuously increased until the torque command reaches a maximum torque setpoint or until the motion system accelerates to a first velocity checkpoint (e.g., velocity checkpoint v3 of FIG. 5) in response to the applied torque command. The maximum torque setpoint and first velocity checkpoint can correspond to upper operational bounds for the motion system, and can be set prior to testing (e.g., a maximum torque and velocity expected during normal operation of the motion system). The rate at which the torque command is increased can also be defined by the user. In one or more embodiments, if the torque setpoint reaches the maximum torque setpoint before the velocity of the motion system reaches the first velocity checkpoint, the torque command can be held at the maximum torque value until the motion system accelerates to the first velocity checkpoint. If the velocity of the motion system does not reach the first velocity checkpoint within a defined timeout period, an appropriate timeout handling sequence can be initiated.

**[00100]** As the motion system accelerates from rest toward the first velocity checkpoint, the velocity will pass through a second velocity checkpoint (e.g., velocity checkpoint v2 of FIG. 5), where the second velocity checkpoint is greater than zero and less than the first velocity checkpoint. The acceleration phase of the test is initiated when the velocity initially reaches this second velocity checkpoint.

**[00101]** Upon determining that the motion system has accelerated to the first velocity checkpoint, the torque command can be continuously decreased at 904, until the torque command reaches a minimum torque setpoint or until the motion system decelerates back to the first velocity checkpoint. In this regard, since the motion system was accelerating at the time the first velocity checkpoint was reached at step 1002, the velocity will continue to increase beyond the first velocity checkpoint for some time after the torque signal begins decreasing in step 1004. The decreasing torque command signal will subsequently cause the motion system to decelerate back to the first velocity

## LSTP102WO

checkpoint. As in step 1002, the rate at which the torque command is decreased can be configured as a user-defined parameter. In the present example, the torque command decreases to zero prior to the motion system returns to the first velocity checkpoint, and continues to decrease in the negative direction until the minimum torque setpoint is reached or until the motion system decelerates to the first velocity checkpoint. That is, the torque command crosses zero during this phase of the testing sequence. This signals the end of the acceleration phase and the beginning of the deceleration phase of the test. Similar to step 1002, if the torque command decreases to the minimum torque setpoint before the motion system reaches the first velocity checkpoint, the torque command will be held at the minimum torque value until the first velocity checkpoint is reached.

**[00102]** When the motion system velocity has returned to the first velocity checkpoint, the torque command value at the time the first velocity checkpoint was reached is maintained as a constant value at 1006 until the motion system decelerates back to the second velocity checkpoint. This triggers the end of the deceleration phase of the test.

**[00103]** At 1008, the time at which the torque command crossed zero during step 1004 is determined. This time, designated  $T_{CROSSOVER}$ , can be used to demarcate the acceleration phase and deceleration phase of the test sequence (in the example described above in connection with FIG. 4,  $T_{CROSSOVER} = t_5$ ). At 1010, integrations are performed on the acceleration phase portions of the torque command curve and corresponding velocity curve of the motion system. That is, the torque command data is integrated from time  $T_0$  to  $T_{CROSSOVER}$ , where  $T_0$  represents the start time for the acceleration phase (the time at which the velocity initially crossed the second velocity checkpoint; e.g., time  $t_3$  of FIG. 4). The result of this acceleration phase integration of the torque command is designated as  $U_{acc}$ . Likewise, the continuous velocity data measured from the motion control system in response to the applied torque command is integrated from time  $T_0$  to  $T_{CROSSOVER}$  to yield an integrated velocity result  $V_{acc}$  for the acceleration phase.

**[00104]** At 1012, similar integrations are performed for the deceleration portions of the torque and velocity data. That is, the torque and velocity data

LSTP102WO

are integrated from time  $T_{CROSSOVER}$  to  $T_{FINAL}$ , where  $T_{FINAL}$  is the end time for the deceleration phase, corresponding to the time at which the motion system has decelerated back to the second velocity checkpoint at step 1006 (in the example described above in connection with FIG. 4,  $T_{FINAL} = t7$ ). The results of these deceleration phase integrations for the torque and velocity data are designated as  $U_{dec}$  and  $V_{dec}$ , respectively.

**[00105]** At 1014, the estimated inertia and/or friction coefficient for the motion system is calculated based on the integrals  $U_{acc}$ ,  $V_{acc}$ ,  $U_{dec}$ , and  $V_{dec}$ . For example, the estimated inertia and friction coefficient may be calculated based on equations (12) and (13), respectively, or a variation thereof.

**[00106]      Motion Profile Generator**

**[00107]** FIG. 11 is a block diagram of an exemplary non-limiting motion profile generating system capable of generating motion profiles for point-to-point moves of a motion control system. Motion profile generating system 1102 can include a position profile generator 1104, a velocity profile generator 1106, an interface component 1108, one or more processors 1110, and memory 1112. In various embodiments, one or more of the position profile generator 1104, velocity profile generator 1106, interface component 1108, the one or more processors 1110, and memory 1112 can be electrically and/or communicatively coupled to one another to perform one or more of the functions of the motion profile generating system 1102. In some embodiments, components 1104, 1106, and 1108 can comprise software instructions stored on memory 1112 and executed by processor(s) 1110. The motion profile generating system 1102 may also interact with other hardware and/or software components not depicted in FIG. 11. For example, processor(s) 1110 may interact with one or more external user interface devices, such as a keyboard, a mouse, a display monitor, a touchscreen, or other such interface devices.

**[00108]** Interface component 1108 can be configured to receive user input and to render output to the user in any suitable format (e.g., visual, audio, tactile, etc.). User input can be, for example, user-entered constraints (e.g., maximum acceleration, maximum velocity, etc.) used by the motion

## LSTP102WO

profile generating system 1102 to calculate a motion profile (to be described in more detail below). Position profile generator 1104 can be configured to receive an indication of a desired target position for a motion system and calculate a motion profile for transitioning to the target position within the parameters of the user-defined constraints. Similarly, velocity profile component 1106 can receive an indication of a desired target velocity for the motion control system and generate a motion profile for transitioning the motion system from a current velocity to the target velocity in conformance with the defined constraints. While FIG. 11 depicts the motion profile generating system as including both the position profile generator 1104 and the velocity profile generator 1106, it is to be appreciated that some embodiments of the motion profile generating system 1102 may include only one of the position profile generator 1104 or the velocity profile generator 1106 without deviating from the scope of this disclosure. The one or more processors 1110 can perform one or more of the functions described herein with reference to the systems and/or methods disclosed. Memory 1112 can be a computer-readable storage medium storing computer-executable instructions and/or information for performing the functions described herein with reference to the systems and/or methods disclosed.

**[00109]** In some embodiments, the profile generator described herein can be an integrated component of a motion controller. FIG. 12 illustrates an exemplary motion control system 1200 comprising a master controller 1202 that utilizes a profile generator 1206 according to one or more embodiments of this disclosure. Master controller 1202 can be, for example, a programmable logic controller (PLC) or other such controller that monitors and controls a system (e.g., an industrial process, an automation system, a batch process, *etc.*) that includes one or more motion devices. In this example, profile generator 1206 can be a functional component of the controller's operating system and/or control software executed by one or more processors residing on the controller 1202. Profile generator 1206 can also be a hardware component residing within controller 1202, such as a circuit board or integrated circuit, that exchanges data with other functional elements of the controller 1202. Other suitable implementations of profile generator 1206 are also within the scope of certain embodiments of this disclosure. For example,

## LSTP102WO

although profile generator 1206 is illustrated in FIG. 12 as being an integrated component of controller 1202, the profile generator 1206 may be a separate element from controller 1202 in some embodiments. For such configurations, profile generator 1206 can exchange data with controller 1202 or other elements of the motion system via any suitable communications means, including but not limited to wired or wireless networking, hardwired data links, or other such communication means.

**[00110]** Exemplary motion control system 1200 also comprises a motor drive 1222, which includes a motion controller 1214 for controlling a motion device (e.g., a motor, not shown) in accordance with a motion profile 1212 provided by master controller 1202. The motion profile 1212 defines a trajectory for transitioning the motion device from a current position or velocity to a target position or velocity, where the trajectory is defined in terms of one or more of a position reference, a velocity reference, an acceleration reference, and/or a jerk reference. In response to receiving motion profile data from master controller 1202, motor controller 1214 will translate the motion profile 1212 into control signaling 1216, which is sent to the motion device to effect transitioning of the motion device to the target position or velocity. If the motor controller 1214 is a closed-loop controller, motor controller 1214 will also monitor a feedback signal 1220 indicating an actual state of the motion device (e.g., the real-time position, velocity, *etc.*) as the control signaling 1216 is being applied. Based on this feedback signal 1220, the motor controller 1214 will adjust the control signaling 1216 as necessary to ensure that the motion device moves in accordance with the motion profile 1212 as closely as possible. Alternatively, if the motor controller 1214 is an open-loop controller, the motor controller 1214 will still generate control signaling 1216 based on motion profile 1212, but will not monitor the feedback signal 1220 during the resulting move.

**[00111]** In the present example, master controller 1202 controls the system in accordance with a control program 1210, which is stored and executed on the controller 1202. During operation, control program 1210 may require that the motion device move to a new position, or transition to a new velocity. The destination position or velocity 1208 is provided to profile generator 1206, which calculates a motion profile 1212 that defines a



## LSTP102WO

trajectory for the move. Profile generator 1206 calculates the motion profile 1212 as a function of one or more motion constraints 1204, which can represent mechanical constraints of the motion system or user preferences regarding operation of the motion device. Motion constraints 1204 can be provided by the user prior to operation (e.g., via interface component 1108 of FIG. 11). In some embodiments, profile generator 1206 can also calculate the motion profile 1212 based additionally on the sample time 1218 of the controller 1202, to ensure that the profile segments align with the controller's sample points, as will be discussed in more detail below.

**[00112]** As will be described in more detail below, motion profile 1212 can define the trajectory of the point-to-point move over time in terms of one or more of a position reference, a velocity reference, an acceleration reference, and a jerk reference. These references represent functions calculated by the motion profile generator 1206 defining how the respective motion attributes will be controlled as a function of time for a given point-to-point move. These references are mathematically related to one another as derivatives. That is, jerk is the derivative of acceleration, acceleration is the derivative of velocity, and velocity is the derivative of position. Profile generator 1206 can calculate these references for respective stages of the trajectory profile, as will be discussed in more detail below.

**[00113]** Once the motion profile 1212 for the move is calculated, profile generator 1206 provides the motion profile 1212 to the motor controller 1214, which translates the motion profile 1212 into control signaling 1216 that instructs the motion device to perform the desired point-to-point move in accordance with the motion profile 1212. As described above, if the motor controller 1214 is a closed-loop controller, control signaling 1216 will be a function of the motion profile 1212 as well as feedback signal 1220, which informs the motor controller 1214 of the actual state of the motion device in real-time. For open-loop control systems, the control signaling 1216 will be a function only of the motion profile 1212.

**[00114]** It is to be understood that the architecture depicted in FIG. 122 is only intended to be an exemplary context in which profile generator 1206 may operate, and that other operating contexts are within the scope of this disclosure. For example, in some scenarios, master controller 1202 may be a

## LSTP102WO

self-contained controller that includes integrated motor control capabilities. In such applications, the controller 1202 may itself translate the motion profile 1212 into a suitable control signal 316 and send this control signal 1216 to the motion device, rather than providing the motion profile 1212 to a separate motor drive 1222. In another exemplary architecture, profile generator 1206 may be an integrated component of motor drive 1222.

**[00115]** Profile generator 1206 can be one or both of a position profile generator or a velocity profile generator. These two types of profile generators are illustrated in FIGs. 13 and 14, respectively. As shown in FIG. 13, position profile generator 1302 receives as inputs a set of constraints 1304, which can represent mechanical constraints of the controlled system or user preferences regarding behavior of the motion system. These constraints can include upper limits on the velocity, acceleration, deceleration, and jerk, as well as a sample time representing an update period of the controller's control signal (typically measured in milliseconds). These constraint values may be set by the user (e.g., via interface component 1108 of FIG. 11), although in some embodiments the position profile generator 1302 may determine the controller's sample time automatically. These constraints 1304 may be set once during deployment of the motion control system, or may be reconfigured for each move. Position profile generator 1302 allows the acceleration and deceleration limits to be configured individually to accommodate profiles having asymmetrical acceleration and deceleration. The sample time is used by the profile generator 1302 to improve accuracy of the motion profile, as will be described in more detail below.

**[00116]** During operation, the position profile generator 1302 will receive a position step command 1308 specifying a new target position for the motion system. Position step command 1308 may be generated by the control program executing on the controller (e.g., control program 1210 of FIG. 12), or may be a move instruction manually input by a user. In response to the position step command 1308, position profile generator calculates a constraint-based, time-optimal motion profile 1306 defining a trajectory for moving the load from its current position to the target position defined by the position step command 1308. The motion profile 1306 comprises one or more of a jerk reference, an acceleration reference, a velocity reference, or a

## LSTP102WO

position reference (which are mathematically related to each other as derivatives). Position profile generator 1302 defines these references as functions of time for each of a set of defined motion profile stages or segments. Table 1 summarizes the seven segments of a point-to-point motion profile.

No.	Segment Name	Description
1	ACC_INC	Acceleration profile increases from zero to maximum acceleration
2	ACC_HOLD	Acceleration profile stays constant as the maximum acceleration
3	ACC_DEC	Acceleration profile decreases from maximum acceleration to zero
4	VEL_HOLD	Velocity profile stays constant
5	DEC_INC	Acceleration profile absolute value increases from zero to maximum deceleration
6	DEC_HOLD	Acceleration profile absolute value stays constant as the maximum deceleration
7	DEC_DEC	Acceleration profile absolute value decreases from maximum deceleration to zero

**Table 1**

**[00117]** Initially, during the first stage (ACC\_INC), the acceleration increases continuously from zero to a constant acceleration. In some scenarios, this constant acceleration will be the maximum acceleration defined by constraints 1304. However, for relatively short position steps this the position profile generator 1302 may determine that a smaller acceleration would result in a more accurate transition to the target position. During the second stage (ACC\_HOLD), the acceleration is held at the constant rate. As the system approaches the target velocity calculated by the position profile generator 1302, the third stage (ACC\_DEC) is entered, during which the acceleration is gradually decreased until the constant velocity is reached. When the constant velocity has been achieved, this constant velocity is held during the fourth stage (VEL\_HOLD) as the system approaches the target position. When the system is near the target position, the trajectory enters the



## LSTP102WO

fifth stage (DEC\_INC), during which the system begins decelerating at a gradually increasing rate from zero to a target deceleration defined by the motion profile. When the target deceleration is reached, this deceleration is held during the sixth stage (DEC\_HOLD). Finally, during the seventh stage (DEC\_DEC), the deceleration is gradually decreased until the system reaches zero velocity, ending the move sequence.

**[00118]** When provided with a position step command 1308, position profile generator 1302 determines which of these seven profile segments are required for a time-optimal motion profile, and calculates one or more of a time varying jerk reference, acceleration reference, velocity reference, or position reference for each segment deemed necessary for the move. The calculated references for the respective stages are combined to yield a complete motion profile, which can be used by an open-loop or closed-loop motion controller (e.g., a motor drive) to drive the motion system through the trajectory defined by the motion profile.

**[00119]** FIG. 14 illustrates an exemplary velocity profile generator 1402 according to one or more embodiments. Velocity profile generator 1402 is similar to position profile generator 1302, but is used to calculate motion profiles in response to a desired change in velocity rather than a change in position. That is, velocity profile generator 1402 calculates a time-optimal motion profile 1406 for transitioning a motion system from a current velocity to a target velocity specified by velocity setpoint 1408. Since transition to a desired velocity setpoint is typically indifferent to the motion system's position, the constraints 1404 defined for the velocity profile generator 1402 may omit the position limit. Likewise, the motion profile 1406 generated by velocity profile generator 1402 may omit a position reference, and define the motion profile exclusively in terms of a time varying jerk reference, acceleration reference and/or velocity reference.

**[00120]** In some motion control applications, motion controllers generate one of either trapezoidal motion profiles or S-curve motion profiles. In addition to or instead of these profile types, the profile generator of the present disclosure can generate profiles according to a third profile type, referred to herein as an ST-curve profile. FIG. 15 compares an exemplary ST-curve profile with traditional trapezoidal and S-curve profiles. The time

LSTP102WO

graphs illustrated in FIG. 15 plot the position, velocity, acceleration, and jerk for a given motion trajectory between position 0 (the start position) and position 2.5 (the target position, as may be defined by position step command 1308 of FIG. 13). As is generally understood, the plotted values are mathematically related to one another as derivatives. That is, velocity is the derivative of position (*i.e.*, the rate of change of position), acceleration is the derivative of velocity, and jerk is the derivative of acceleration.

**[00121]** Trapezoidal motion profiles only employ three of the seven profile stages described above – constant acceleration (stage 2), constant velocity (stage 4), and constant deceleration (stage 6). This results in the trapezoidal velocity profile depicted by the dotted line of the velocity curve in FIG. 15. The abrupt transitions between the constant acceleration/deceleration stages and the constant velocity stage results in sharp corners at the top of the trapezoidal velocity curve. Since the acceleration and deceleration phases of the trapezoidal profile are always constant, the acceleration curve for this profile steps abruptly between constant values, as illustrated by the dotted line on the acceleration graph. In the present example, the rate of deceleration is twice that of acceleration, so the acceleration curve for the trapezoidal case steps to 0.5 during the acceleration stage, 0 for the constant velocity stage, and -1.0 for the deceleration stage. Also, the jerk curve (representing the rate of change of acceleration/deceleration) pulses briefly during moments of transition (not plotted) and remains at zero when acceleration or deceleration remains constant, as shown by the dotted line on the jerk graph of FIG. 15.

**[00122]** Since the trapezoidal profile always accelerates and decelerates at a constant rate without gradual transitioning to and from the constant velocity stage, the trapezoidal curve profile can traverse the distance between the current position and the target position relatively quickly. However, the sudden transitions between acceleration/deceleration and constant (or zero) velocity can introduce undesirable mechanical turbulence in the system. Additionally, since the deceleration does not decrease gradually as the motion system approaches the target position, but instead maintains constant deceleration until the target position is reached before suddenly shifting to zero velocity, the trapezoidal motion profile has a high likelihood of

## LSTP102WO

overshooting the target position at the end of the initial traversal, requiring the controller to apply a compensatory control signal to bring the load back to the target position. This process may need to be iterated several times before the system settles on the target position, introducing undesirable system oscillations.

**[00123]** In contrast to the trapezoidal profile, the S-curve profile (depicted as the light solid line in the graphs of FIG. 15) utilizes all the continuous acceleration/deceleration stages of the seven profile stages, thereby allowing gradual transitions between the constant (or zero) velocity phases and the constant acceleration/deceleration stages. These gradual acceleration transitions can be seen clearly on the acceleration graph. Rather than starting at a constant acceleration from time 0, as does the trapezoidal profile, the acceleration for the S-curve profile ramps gradually to constant acceleration starting at time 0. When the velocity has reached maximum (between 1s and 2s), the acceleration gradually decreases to zero to achieve constant velocity, rather than stepping down abruptly to zero as in the trapezoidal case. Similar gradual changes in acceleration can be seen during the later deceleration stages for the S-curve profile. The effect of these gradual acceleration changes on the velocity and position curves can be seen on the respective velocity and position graphs. In particular, the corners of the S-curve velocity profile are rounder relative to those of the trapezoidal curve, representing a smoother transition between the acceleration/deceleration stages and the constant velocity stages. Likewise, the S-curve position profile shows a smoother transition between the initial position and the target position, though at the expense of additional time required to reach the target.

**[00124]** One or more embodiments of the profile generator described herein can support generation of S-curve motion profiles. Conventionally, motion control systems that utilize S-curve motion profiles only support symmetrical acceleration and deceleration; that is, the absolute values of the constant acceleration and the constant deceleration stages are equal. By contrast, the profile generator described herein can support S-curve motion profiles having asymmetrical acceleration and deceleration. This is illustrated on the acceleration graph of FIG. 15, which depicts the S-curve as having a

## LSTP102WO

limit of 0.5 during acceleration, and a limit of -1 during deceleration. To provide for such asymmetric acceleration and deceleration, the profile generator can allow separate acceleration and deceleration limits to be configured as system constraints (see, e.g., constraints 1304 of FIG. 13), and calculate the motion profile in view of these constraints.

**[00125]** As illustrated on the jerk graph, the rate at which the acceleration/deceleration increases and decreases during stages 1, 3, 5, and 7 of the motion profile for the S-curve case are always constant. That is, the jerk is always a constant value for any given stage of the motion profile – either 1, 0, or -1 in the present example. This can result in sharp transitions between the increasing/decreasing acceleration (or deceleration) stages and constant acceleration stages, as illustrated on the acceleration graph.

**[00126]** To facilitate smoother motion than that offered by the trapezoidal and S-curve profiles of conventional motion control systems while achieving time-optimal transition between positions, one or more embodiments of the profile generator described herein can calculate motion profiles that accord to the ST-curve profile type. An exemplary ST-curve profile is represented as the dark solid line on the graphs of FIG. 15. In contrast to the trapezoidal and S-curve profiles, the ST-curve profile gradually varies the jerk continuously over time during the stages of increasing and decreasing acceleration and deceleration. This can result in the smoother acceleration transitions illustrated on the acceleration graph of FIG. 15, and the corresponding smoother velocity and position curves shown in the respective velocity and position graphs.

**[00127]** Moreover, ST-curve profiles can support asymmetrical acceleration and deceleration (that is, the profile generator can calculate profiles having rates of acceleration that differ from the rates of deceleration for a given motion profile). Deriving a mathematical trajectory expression as a function of time while simultaneously finding a time-optimal solution can be challenging when using asymmetric acceleration/deceleration. To address this concern, one or more embodiments of the profile generator described herein can employ an algorithm that leverages a relationship between acceleration and deceleration, and between acceleration jerk and deceleration jerk, and substitute these relationships during the derivation, making it

LSTP102WO

possible to derive the analytical expressions of the trajectories and then find the time-optimal solution.

**[00128]** An exemplary ST-curve position profile is derived below. One or more embodiments of the profile generator described herein can generate motion profile references based on the following derivations. However, it is to be understood that the profile generator described herein is not limited to this technique for generating motion profiles based on ST-curves, and that any suitable algorithm that yields a continuous jerk curve defined as a function of time is within the scope of this disclosure.

**[00129]** In the following equations,  $\ddot{\theta}$ ,  $\dot{\theta}$ ,  $\dot{\theta}$ , and  $\theta$  are jerk, acceleration, velocity, and position, respectively.  $t_1$ ,  $t_2$ ,  $t_3$ ,  $t_4$  and  $t_5$  are the respective durations of the ACC\_INC, ACC\_HOLD, VEL\_HOLD, DEC\_INC, and DEC\_HOLD stages of the motion profile (see Table 1 above). In the present example, it is assumed that ACC\_INC and ACC\_DEC are equal in duration, and thus  $t_1$  is the duration of both the ACC\_INC and ACC\_DEC stages. Likewise, DEC\_INC is assumed to be equal in duration to DEC\_DEC, so  $t_4$  is the duration for both DEC\_INC and DEC\_DEC.  $K$  is a gain value to be determined for each stage of the motion profile for each of the jerk, acceleration, velocity, and position, according to the following equations (where, for each of the seven stages,  $t = 0$  represents the start time of the respective stage):

$$\ddot{\theta}(t) = K \begin{cases} t(t_1 - t) & 0 \leq t \leq t_1 \\ 0 & 0 \leq t \leq t_2 \\ -t(t_1 - t) & 0 \leq t \leq t_1 \\ 0 & 0 \leq t \leq t_3 \\ -t(t_4 - t) & 0 \leq t \leq t_4 \\ 0 & 0 \leq t \leq t_5 \\ t(t_4 - t) & 0 \leq t \leq t_4 \end{cases} \quad (14)$$

LSTP102WO

$$\ddot{\theta}(t) = K \begin{cases} t^2 \frac{3t_1 - 2t}{6} & 0 \leq t \leq t_1 \\ \frac{t_1^3}{6} & 0 \leq t \leq t_2 \\ \frac{t_1^3}{6} + t^2 \frac{2t - 3t_1}{6} & 0 \leq t \leq t_1 \\ 0 & 0 \leq t \leq t_3 \\ t^2 \frac{2t - 3t_4}{6} & 0 \leq t \leq t_4 \\ -\frac{t_4^3}{6} & 0 \leq t \leq t_5 \\ -\frac{t_4^3}{6} + t^2 \frac{3t_4 - 2t}{6} & 0 \leq t \leq t_4 \end{cases} \quad (15)$$

$$\dot{\theta}(t) = K \begin{cases} \frac{t^3}{12}(2t_1 - t) & 0 \leq t \leq t_1 \\ \frac{t_1^3(2t + t_1)}{12} & 0 \leq t \leq t_2 \\ \frac{t_1^3(t_1 + t_2)}{6} - \frac{1}{12}(t + t_1)(t_1 - t)^3 & 0 \leq t \leq t_1 \\ \frac{t_1^3(t_1 + t_2)}{6} & 0 \leq t \leq t_3 \\ \frac{t_1^3(t_1 + t_2)}{6} - \frac{t^3}{12}(2t_4 - t) & 0 \leq t \leq t_4 \\ \frac{t_1^3(t_1 + t_2)}{6} - \frac{t_4^3}{12}(2t + t_4) & 0 \leq t \leq t_5 \\ \frac{t_1^3(t_1 + t_2)}{6} - \frac{t_4^3(t_4 + t_5)}{6} + \frac{(t_4 - t)^3(t + t_4)}{12} & 0 \leq t \leq t_4 \end{cases} \quad (16)$$

LSTP102WO

$$\theta(t) = K \begin{cases} \frac{t^4}{120}(5t_1 - 2t) & 0 \leq t \leq t_1 \\ \frac{t_1^5}{40} + \frac{t_1^3(t+t_1)}{12} & 0 \leq t \leq t_2 \\ \frac{t_1^3}{12}(t_1+t_2)(2t+t_2) + \frac{2t+3t_1}{120}(t_1-t)^4 & 0 \leq t \leq t_1 \\ \frac{t_1^3(t_1+t_2)}{12}(2t+2t_1+t_2) & 0 \leq t \leq t_3 \\ \frac{t_1^3(t_1+t_2)}{12}(2t+2t_3+2t_1+t_2) + \frac{t^5}{60} - \frac{t_4 t^4}{24} & 0 \leq t \leq t_4 \\ \frac{t_1^3(t_1+t_2)}{12}(2t+2t_4+2t_3+2t_1+t_2) - \frac{t_4^5}{40} - \frac{t_4^3 t^2}{12} - \frac{t_4^4 t}{12} & 0 \leq t \leq t_5 \\ \frac{t_1^3(t_1+t_2)}{12}(t_5+2t_4+2t_3+2t_1+t_2) - \frac{(t_4-t)^4(2t+3t_4)}{120} & 0 \leq t \leq t_4 \end{cases} \quad (17)$$

**[00130]** Given these relationships, the maximum acceleration jerk, maximum deceleration jerk, maximum acceleration, maximum deceleration, and maximum velocity can be described in terms of the segment durations:

$$K = P \frac{12}{t_1^3(t_1+t_2)(t_5+2t_4+2t_3+2t_1+t_2)} \quad (18)$$

$$J = P \frac{3}{t_1(t_1+t_2)(t_5+2t_4+2t_3+2t_1+t_2)} \quad (19)$$

$$I = P \frac{3}{t_4(t_4+t_5)(t_5+2t_4+2t_3+2t_1+t_2)} \quad (20)$$

$$A = P \frac{2}{(t_1+t_2)(t_5+2t_4+2t_3+2t_1+t_2)} \quad (21)$$

$$D = P \frac{2}{(t_4+t_5)(t_5+2t_4+2t_3+2t_1+t_2)} \quad (22)$$

$$V = P \frac{2}{(t_5+2t_4+2t_3+2t_1+t_2)} \quad (23)$$

where:

LSTP102WO

$P$  is the position step,

$J$  is the maximum acceleration jerk,

$I$  is the maximum deceleration jerk,

$A$  is the maximum acceleration,

$D$  is the maximum deceleration, and

$V$  is the maximum velocity.

**[00131]** The relationships among  $P$ ,  $V$ ,  $A$ ,  $D$ ,  $J$ ,  $I$ ,  $t_1$ ,  $t_2$ ,  $t_3$ ,  $t_4$ , and  $t_5$  can now be obtained:

$$t_1^3(t_1 + t_2) = t_4^3(t_4 + t_5) \quad (24)$$

$$t_1 = \frac{3A}{2J} \quad (25)$$

$$t_4 = \frac{3D}{2I} \quad (26)$$

$$t_2 = \frac{V}{A} - \frac{3A}{2J} \quad (27)$$

$$t_5 = \frac{V}{D} - \frac{3D}{2I} \quad (28)$$

$$t_3 = \frac{P}{V} - \frac{3A}{4J} - \frac{V}{2A} - \frac{3D}{4I} - \frac{V}{2D} \quad (29)$$

$$\frac{A^2}{D^2} = \frac{J^3}{I^3} \quad (30)$$

**[00132]** Given that  $t_1$ ,  $t_2$ ,  $t_3$ ,  $t_4$ , and  $t_5$  should all be greater than or equal to zero, and assuming that

$$\frac{D}{A} = \rho \quad (31)$$



LSTP102WO

$$\frac{I}{J} = \rho^{\frac{2}{3}} \quad (32)$$

the following set of inequalities can be established:

$$V \geq \frac{3A^2}{2J} = \frac{3}{2I\rho^{\frac{4}{3}}} D^2 \quad (33)$$

$$V \geq \frac{3}{2I} D^2 \quad (34)$$

$$\frac{P}{V} \geq \frac{3}{4I} \rho^{-\frac{1}{3}} D + \frac{\rho V}{2D} + \frac{3}{4I} D + \frac{1}{2} \frac{V}{D} \quad (35)$$

$$\frac{P}{V} \geq \left( \rho^{-\frac{1}{3}} + 1 \right) \frac{3}{4I} D + \frac{1}{2} (\rho + 1) \frac{V}{D} \quad (36)$$

**[00133]** Solving inequalities (33)-(36) yields appropriate values for  $V$ ,  $A$ ,  $D$ ,  $J$ , and  $I$  (the maximum values for velocity, acceleration, deceleration, acceleration jerk, and deceleration jerk, respectively).

**[00134]** Substituting these maximum values into equations (25)-(29) can yield values for  $t_1$ ,  $t_2$ ,  $t_3$ ,  $t_4$ , and  $t_5$  (the durations of the respective segments of the motion profile). The values for  $V$ ,  $A$ ,  $D$ ,  $J$ ,  $I$ ,  $t_1$ ,  $t_2$ ,  $t_3$ ,  $t_4$ , and  $t_5$  derived according to equations (14) – (36) above can produce a smooth, time-optimal trajectory that operates within the defined mechanical constraints or user demands.

**[00135]** Based on the relationships described above, the profile generator can calculate a suitable ST-curve motion profile for a given point-to-point move. It is recognized, however, that the values initially calculated for  $t_1$ ,  $t_2$ ,  $t_3$ ,  $t_4$ , and  $t_5$  may not be multiples of the controller's sample time, and consequently may not align with the sample points of the motion controller. When a profile segment duration falls between two controller sample points, it may be necessary for the controller to compensate for small differences between the desired control signal output and the actual control signal output. To address this issue, one or more embodiments of the profile generator

## LSTP102WO

described herein can perform an additional computation after the maximum values  $V$ ,  $A$ ,  $D$ ,  $J$ , and  $I$  and the segment durations  $t_1$ ,  $t_2$ ,  $t_3$ ,  $t_4$ , and  $t_5$  have been derived as described above.

**[00136]** Specifically, after the profile generator has calculated  $t_1$ ,  $t_2$ ,  $t_3$ ,  $t_4$ , and  $t_5$  according to the above derivations, each of these duration values can be upper-rounded to the nearest sample time to yield  $t_{1New}$ ,  $t_{2New}$ ,  $t_{3New}$ ,  $t_{4New}$ , and  $t_{5New}$ . This rounding step can be based on the sample time provided to the profile generator as one of the constraints 1304 or 1404. The profile generator can then calculate new values for  $V$ ,  $A$ ,  $D$ ,  $J$ , and  $I$  using the rounded duration values  $t_{1New}$ ,  $t_{2New}$ ,  $t_{3New}$ ,  $t_{4New}$ , and  $t_{5New}$ . This recalculation yields a final motion profile comprising segment durations that are multiples of the sample time, which can ensure that the control signal output by the controller is aligned with the controller's sample points, thereby mitigating the need to compensate for the small difference introduced when the motion profile times fall between two sample points.

**[00137]** Alternatively or in addition to the ST-curves described above, one or more embodiments of the profile generator described herein is capable of generating S-curve profiles having asymmetric acceleration and deceleration (see, *e.g.*, the exemplary S-curve trajectory of FIG. 15). An exemplary S-curve profile having asymmetric acceleration and deceleration is derived below. One or more embodiments of the profile generator described herein can generate motion profile references based on the following derivations or variants thereof.

**[00138]** As in the ST-curve equations derived above,  $\ddot{\theta}$ ,  $\ddot{\theta}$ ,  $\dot{\theta}$ , and  $\theta$  are jerk, acceleration, velocity, and position, respectively.  $t_1$ ,  $t_2$ ,  $t_3$ ,  $t_4$  and  $t_5$  are the respective durations of the ACC\_INC, ACC\_HOLD, VEL\_HOLD, DEC\_INC, and DEC\_HOLD stages of the motion profile (see Table 1 above). As in the ST-curve example, it is assumed that ACC\_INC and ACC\_DEC are equal in duration, and thus  $t_1$  is the duration of both the ACC\_INC and ACC\_DEC stages. Likewise, DEC\_INC is assumed to be equal in duration to DEC\_DEC, so  $t_4$  is the duration for both DEC\_INC and DEC\_DEC.  $K$  is a gain value to be determined for each stage of the motion profile for each of the jerk, acceleration, velocity, and position, according to the following

## LSTP102WO

equations (where, for each of the seven stages,  $t = 0$  represents the start time of the respective stage):

$$\ddot{\theta}(t) = K \begin{cases} 1, & 0 < t < t_1 \\ 0, & 0 < t < t_2 \\ -1, & 0 < t < t_1 \\ 0, & 0 < t < t_3 \\ -1, & 0 < t < t_4 \\ 0, & 0 < t < t_5 \\ 1, & 0 < t < t_4 \end{cases} \quad (37)$$

$$\ddot{\theta}(t) = K \begin{cases} t, & 0 < t < t_1 \\ t_1, & 0 < t < t_2 \\ t_1 - t, & 0 < t < t_1 \\ 0, & 0 < t < t_3 \\ -t, & 0 < t < t_4 \\ -t_4, & 0 < t < t_5 \\ t - t_4, & 0 < t < t_4 \end{cases} \quad (38)$$

$$\dot{\theta}(t) = K \begin{cases} \frac{1}{2}t^2, & 0 < t < t_1 \\ t_1 \left( t + \frac{1}{2}t_1 \right), & 0 < t < t_2 \\ \frac{t_1}{2}(2t_2 + t_1) + t_1 t - \frac{1}{2}t^2, & 0 < t < t_1 \\ t_1(t_2 + t_1), & 0 < t < t_3 \\ t_1(t_2 + t_1) - \frac{1}{2}t^2, & 0 < t < t_4 \\ t_1(t_2 + t_1) - \frac{1}{2}t_4^2 - t_4 t, & 0 < t < t_5 \\ t_1(t_2 + t_1) - t_4(t_4 + t_5) + \frac{1}{2}(t_4 - t)^2, & 0 < t < t_4 \end{cases} \quad (39)$$

LSTP102WO

$$\theta(t) = K \begin{cases} \frac{1}{6}t^3, & 0 < t < t_1 \\ \frac{1}{6}t_1^3 + \frac{1}{2}t_1(t+t_1)t, & 0 < t < t_2 \\ \frac{1}{2}t_1(t_2+t_1)(t_2+2t) + \frac{1}{6}(t_1-t)^3, & 0 < t < t_1 \\ \frac{1}{2}t_1(t_1+t_2)(2t_1+t_2+2t), & 0 < t < t_3 \\ \frac{1}{2}t_1(t_1+t_2)(2t_1+t_2+2t_3+2t) - \frac{1}{6}t^3, & 0 < t < t_4 \\ \frac{1}{2}t_1(t_1+t_2)(2t_1+t_2+2t_3+2t_4+2t) - \frac{1}{6}t_4^3 - \frac{1}{2}t_4^2t - \frac{1}{2}t_4t^2, & 0 < t < t_5 \\ \frac{1}{2}t_1(t_1+t_2)(2t_1+t_2+2t_3+2t_4+t_5) - \frac{1}{6}(t_4-t)^3, & 0 < t < t_4 \end{cases} \quad (40)$$

**[00139]** Given these relationships, the maximum acceleration jerk, maximum deceleration jerk, maximum acceleration, maximum deceleration, and maximum velocity can be described in terms of the segment durations:

$$K = P \frac{2}{t_1(t_1+t_2)(t_5+2t_4+2t_3+2t_1+t_2)} \quad (41)$$

$$J = I = K \quad (42)$$

$$A = P \frac{2}{(t_1+t_2)(t_5+2t_4+2t_3+2t_1+t_2)} \quad (43)$$

$$D = P \frac{2}{(t_4+t_5)(t_5+2t_4+2t_3+2t_1+t_2)} \quad (44)$$

$$V = P \frac{2}{t_5+2t_4+2t_3+2t_1+t_2} \quad (45)$$

where:

$P$  is the position step,

$J$  is the maximum acceleration jerk,

LSTP102WO

$J$  is the maximum deceleration jerk,  
 $A$  is the maximum acceleration,  
 $D$  is the maximum deceleration, and  
 $V$  is the maximum velocity.

**[00140]** The relationships among  $P$ ,  $V$ ,  $A$ ,  $D$ ,  $J$ ,  $I$ ,  $t_1$ ,  $t_2$ ,  $t_3$ ,  $t_4$ , and  $t_5$  can now be obtained:

$$t_1(t_1 + t_2) = t_4(t_4 + t_5) \quad (46)$$

$$t_1 = \frac{A}{J} \quad (47)$$

$$t_2 = \frac{V}{A} - \frac{A}{J} \quad (48)$$

$$t_4 = \frac{D}{J} \quad (49)$$

$$t_5 = \frac{V}{D} - \frac{D}{J} \quad (50)$$

$$t_3 = \frac{P}{V} - \frac{A}{2J} - \frac{V}{2A} - \frac{D}{2J} - \frac{V}{2D} \quad (51)$$

**[00141]** Given that  $t_1$ ,  $t_2$ ,  $t_3$ ,  $t_4$ , and  $t_5$  should all be greater than or equal to zero, and assuming that

$$\frac{D}{A} = \rho \quad (52)$$

the following set of inequalities can be established:

$$V \geq \frac{A^2}{J} = \frac{1}{J\rho^2} D^2 \quad (53)$$

LSTP102WO

$$V \geq \frac{1}{J} D^2 \quad (54)$$

$$\frac{P}{V} \geq \frac{1}{2J} \rho^{-1} D + \frac{\rho}{2} \frac{V}{D} + \frac{1}{2J} D + \frac{1}{2} \frac{V}{D} \quad (55)$$

$$\frac{P}{V} \geq (\rho^{-1} + 1) \frac{1}{2J} D + \frac{1}{2} (\rho + 1) \frac{V}{D} \quad (56)$$

**[00142]** Solving inequalities (53)-(56) yields appropriate values for  $V$ ,  $A$ ,  $D$ ,  $J$ , and  $I$  (the maximum values for velocity, acceleration, deceleration, acceleration jerk, and deceleration jerk, respectively) for the S-curve profile.

**[00143]** Substituting these maximum values into equations (47)-(51) can yield values for  $t_1$ ,  $t_2$ ,  $t_3$ ,  $t_4$ , and  $t_5$  (the durations of the respective segments of the S-curve motion profile). The values for  $V$ ,  $A$ ,  $D$ ,  $J$ ,  $I$ ,  $t_1$ ,  $t_2$ ,  $t_3$ ,  $t_4$ , and  $t_5$  derived according to equations (37) – (56) above can produce an S-curve profile having asymmetrical acceleration and deceleration, and that operates within the defined mechanical constraints or user demands.

**[00144]** In some embodiments, the profile generator can adapt the resultant S-curve motion profile to the sample time of the controller by way of an additional calculation similar to that described above in connection with the ST-curve profile. That is, after calculating  $t_1$ ,  $t_2$ ,  $t_3$ ,  $t_4$ , and  $t_5$  according to the above derivations, the profile generator can upper-round these durations to the nearest sample time to yield new duration values  $t_{1New}$ ,  $t_{2New}$ ,  $t_{3New}$ ,  $t_{4New}$ , and  $t_{5New}$ . The profile generator can then calculate new values for  $V$ ,  $A$ ,  $D$ ,  $J$ , and  $I$  using the rounded duration values  $t_{1New}$ ,  $t_{2New}$ ,  $t_{3New}$ ,  $t_{4New}$ , and  $t_{5New}$ .

**[00145]** While motion profiles typically comprise the seven stages listed in Table 1 above, some point-to-point moves may not require all seven segments. For example, if the distance between the current state of the motion system and the target state is relatively small, the VEL\_HOLD (constant velocity) segment of the motion profile may be eliminated. Accordingly, one or more embodiments of the profile generator described herein may support automatic or intelligent segment skipping. That is, rather than perform calculations for all seven stages of the profile, even if one or more of the stages will not be used in the final trajectory, some embodiments

## LSTP102WO

of the profile generator described herein can calculate only those stages that will be used in the final trajectory for a given point-to-point move.

**[00146]** The motion profile can automatically determine which segments can be skipped during calculation of the motion profile when a new move command is received. In some embodiments, the profile generator may determine which segments may be skipped based in part on the total distance between the current position and the target position (in the case of a position change), or the difference between the current velocity and the target velocity (in the case of a velocity change), where smaller differences between the current and target state may suggest elimination of certain segments of the motion profile. In such embodiments, the difference between the current and target states may be compared with a set of defined difference ranges, where each defined difference range is associated with one or more segments that may be omitted from a corresponding motion profile.

**[00147]** One or more embodiments of the profile generator described herein may also infer which segments may be skipped based on historical motion data. For example, the profile generator may record a history of issued move commands and corresponding trajectory data (*e.g.*, position, velocity, acceleration, and/or jerk over time) for the moves performed in response to the commands. The profile generator can analyze this historical data to make an inference regarding which segments may be omitted for a particular type of move. Thus, when a new point-to-point move command is received, prior to calculating the motion profile for the move, the profile generator can infer which segments may be skipped based on the shape of trajectories performed in response to past move commands having similar characteristics (*e.g.*, similar distances to traverse, similar speeds at the time the move command was received, *etc.*).

**[00148]** FIGs. 16-19 illustrate segment skipping for exemplary seven-stage position profiles. FIG. 16 illustrates an exemplary S-curve profile that utilizes all seven stages. FIG. 17 illustrates a profile that skips segment 4 (the constant velocity stage). Profile generator may calculate such a profile in cases for which the position or velocity step to be traversed is small enough that the constant velocity stage will not be reached before the target position or velocity is reached. Upon receipt of such a position step or velocity

LSTP102WO

setpoint command, the profile generator can make this determination prior to performing the profile calculations for the desired move, and will only perform calculations on stages 1-3 and 5-7. Similarly, FIG. 18 illustrates an example profile that skips segments 2 and 6, and FIG. 19 illustrates an example profile that skips segments 2, 4, and 6.

**[00149]** FIGS. 20-22 illustrate various methodologies in accordance with certain disclosed aspects. While, for purposes of simplicity of explanation, the methodologies are shown and described as a series of acts, it is to be understood and appreciated that the disclosed aspects are not limited by the order of acts, as some acts may occur in different orders and/or concurrently with other acts from that shown and described herein. For example, those skilled in the art will understand and appreciate that a methodology can alternatively be represented as a series of interrelated states or events, such as in a state diagram. Moreover, not all illustrated acts may be required to implement a methodology in accordance with certain disclosed aspects. Additionally, it is to be further appreciated that the methodologies disclosed hereinafter and throughout this disclosure are capable of being stored on an article of manufacture to facilitate transporting and transferring such methodologies to computers.

**[00150]** FIG. 20 illustrates an example methodology 2000 for calculating a motion profile for a point-to-point move in a motion control system. At 2002, a set of motion constraints are defined for a motion control system. These constraints can represent physical constraints of a mechanical system controlled by the motion control system, and can include limits on velocity, acceleration, deceleration, and jerk. The constraints can also include a definition of the sample time for the motion controller used to control the mechanical system. At 2002, a command to transition the controlled mechanical system to a new position or velocity is received. This command can originate from a motion control program executed by the motion controller, or can be a manual move command entered by a user. The command can be received by a profile generator associated with the motion controller (e.g., profile generator 1206, 1302, or 1402 described above).

**[00151]** At 2006, in response to receipt of the command received at step 2004, a motion profile can be calculated for moving the mechanical system



## LSTP102WO

from its current position or velocity to the new position or velocity indicated by the command. The profile generator can calculate this motion profile to include a continuous jerk reference defined as a function of time for at least one of the segments of the motion profile. In some embodiments, the motion profile can be calculated as an ST-curve according to the derivations described above in connection with equations (14)-(36). Such a motion profile can yield a jerk reference having the general format depicted by the dark solid line of the jerk graph illustrated in FIG. 15, in which the jerk gradually varies over time between a maximum and minimum value according to the calculated jerk function. At 2008, the controlled mechanical system is instructed to traverse from its current position or velocity to the new position or velocity according to the motion profile defined at step 2006. This can entail, for example, providing the motion profile calculated at step 2006 to a motor drive, which controls a motor that drives the mechanical system in accordance with the motion profile and (in the case of closed-loop control) a feedback signal providing measured real-time state data for the mechanical system.

**[00152]** FIG. 21 illustrates an example methodology 2100 for calculating a constraint-based time-optimal motion profile that conforms to a sample time of a motion controller. At 2102, a set of motion constraints for a controlled mechanical system are defined. As in previous examples, these can include limits on velocity, acceleration, deceleration, and jerk, as well as the sample time of the controller. These constraints can be provided to a profile generator associated with the controller (e.g., profile generator 1206, 1302, or 1402 described above). At 2104, a command to transition the mechanical system from a current position or velocity to a new position or velocity is received (e.g., by the profile generator). At 2106, a motion profile for controlling the trajectory of the mechanical system in response to the command is generated by calculating at least one of the maximum acceleration jerk ( $J$ ), maximum deceleration jerk ( $I$ ), maximum acceleration ( $A$ ), maximum deceleration ( $D$ ), and maximum velocity ( $V$ ) as functions of time for each segment of the motion profile (where the profile can comprise up to seven segments as defined in Table 1 above). In addition, the durations for each segment of the profile are calculated (for example, using the techniques described above in connection with equations (14)-(36)).

LSTP102WO

**[00153]** At 2108, it is determined whether all profile segment durations calculated at step 2106 are multiples of the sample time of the controller. If all segments have durations that are multiples of the sample time, the method moves to step 2114, where a motion profile is generated based on the profile segment durations and the values of  $J$ ,  $I$ ,  $A$ ,  $D$ , and  $V$  calculated at step 2106. Alternatively, if one or more of the profile segments are not an even multiple of the cycle time, the method moves to step 2110, where all profile segment durations are upper-rounded to the nearest multiple of the sample time. At 2112, the values of one or more of  $J$ ,  $I$ ,  $A$ ,  $D$ , and  $V$  are recalculated based on the rounded profile segment durations derived at step 2110. Based on the rounded profile segment durations and the recalculated values of  $J$ ,  $I$ ,  $A$ ,  $D$ , and/or  $V$ , a motion profile is generated at 2114.

**[00154]** FIG. 22 illustrates an example methodology 2200 for efficiently calculating a motion profile for a point-to-point move using segment skipping. At 2202, a command to transition a controlled mechanical system to a new position or new velocity is received (e.g., at profile generator 1206, 1302, or 1402). At 2204, a determination can be made regarding which of the seven segments of the motion profile are required to perform the requested point-to-point move. This determination can be made automatically by the profile generator based, for example, on a determination of the distance that must be traversed between the current position and the desired position (in the case of a position change) or the difference between the current velocity and the desired velocity (in the case of a velocity change).

**[00155]** At 2206, a determination is made as to whether all seven profile segments are required to carry out the desired move, based on the determination made at 2204. If all profile segments are required, the method moves to step 2210, where a motion profile is generated for the point-to-point move by performing profile calculations for all seven segments. Alternatively, if it is determined at step 2206 that one or more profile segments are not required, the method moves to step 2208, where a motion profile is generated for the point-to-point move by performing calculations only for the required segments, as determined at step 2204. Segment skipping according to methodology 2200 can facilitate more efficient calculation of a constraint-

LSTP102WO

based, time-optimal motion profile by reducing unnecessary processing overhead associated with calculating unnecessary profile segments.

#### EXEMPLARY NETWORKED AND DISTRIBUTED ENVIRONMENTS

**[00156]** One of ordinary skill in the art can appreciate that the various embodiments described herein can be implemented in connection with any computer or other client or server device, which can be deployed as part of a computer network or in a distributed computing environment, and can be connected to any kind of data store where media may be found. In this regard, the various embodiments of the video editing system described herein can be implemented in any computer system or environment having any number of memory or storage units (e.g., memory 216 of FIG. 2 or 1112 of FIG. 11), and any number of applications and processes occurring across any number of storage units. This includes, but is not limited to, an environment with server computers and client computers deployed in a network environment or a distributed computing environment, having remote or local storage. For example, with reference to FIG. 2, the torque command generator 204, velocity monitoring component 206, inertia component 208, friction coefficient component 210, and interface component 212 can be stored on a single memory 216 associated with a single device, or can be distributed among multiple memories associated with respective multiple devices. Similarly, torque command generator 204, velocity monitoring component 206, inertia component 208, friction coefficient component 210, and interface component 212 can be executed by a single processor 214, or by multiple distributed processors associated with multiple devices.

**[00157]** Distributed computing provides sharing of computer resources and services by communicative exchange among computing devices and systems. These resources and services include the exchange of information, cache storage and disk storage for objects. These resources and services can also include the sharing of processing power across multiple processing units for load balancing, expansion of resources, specialization of processing, and the like. Distributed computing takes advantage of network connectivity, allowing clients to leverage their collective power to benefit the entire enterprise. In this regard, a variety of devices may have applications, objects

LSTP102WO

or resources that may participate in the various embodiments of this disclosure.

**[00158]** FIG. 23 provides a schematic diagram of an exemplary networked or distributed computing environment. The distributed computing environment includes computing objects 2310, 2312, *etc.* and computing objects or devices 2320, 2322, 2324, 2326, 2328, *etc.*, which may include programs, methods, data stores, programmable logic, *etc.*, as represented by applications 2330, 2332, 2334, 2336, 2338. It can be appreciated that computing objects 2310, 2312, *etc.* and computing objects or devices 2320, 2322, 2324, 2326, 2328, *etc.* may comprise different devices, such as personal digital assistants (PDAs), audio/video devices, mobile phones, MP3 players, personal computers, laptops, tablets, *etc.*, where embodiments of the inertia estimator described herein may reside on or interact with such devices.

**[00159]** Each computing object 2310, 2312, *etc.* and computing objects or devices 2320, 2322, 2324, 2326, 2328, *etc.* can communicate with one or more other computing objects 1110, 1112, *etc.* and computing objects or devices 2320, 2322, 2324, 2326, 2328, *etc.* by way of the communications network 2340, either directly or indirectly. Even though illustrated as a single element in FIG. 23, communications network 2340 may comprise other computing objects and computing devices that provide services to the system of FIG. 23, and/or may represent multiple interconnected networks, which are not shown. Each computing object 2310, 2312, *etc.* or computing objects or devices 2320, 2322, 2324, 2326, 2328, *etc.* can also contain an application, such as applications 2330, 2332, 2334, 2336, 2338 (*e.g.*, inertia estimator 202, motion profile generating system 1102, or components thereof), that might make use of an API, or other object, software, firmware and/or hardware, suitable for communication with or implementation of various embodiments of this disclosure.

**[00160]** There are a variety of systems, components, and network configurations that support distributed computing environments. For example, computing systems can be connected together by wired or wireless systems, by local networks or widely distributed networks. Currently, many networks are coupled to the Internet, which provides an infrastructure for widely distributed computing and encompasses many different networks, though any

LSTP102WO

suitable network infrastructure can be used for exemplary communications made incident to the systems as described in various embodiments herein.

**[00161]** Thus, a host of network topologies and network infrastructures, such as client/server, peer-to-peer, or hybrid architectures, can be utilized. The "client" is a member of a class or group that uses the services of another class or group. A client can be a computer process, *e.g.*, roughly a set of instructions or tasks, that requests a service provided by another program or process. A client process may utilize the requested service without having to "know" all working details about the other program or the service itself.

**[00162]** In a client/server architecture, particularly a networked system, a client can be a computer that accesses shared network resources provided by another computer, *e.g.*, a server. In the illustration of FIG. 23, as a non-limiting example, computing objects or devices 2320, 2322, 2324, 2326, 2328, *etc.* can be thought of as clients and computing objects 2310, 2312, *etc.* can be thought of as servers where computing objects 2310, 2312, *etc.* provide data services, such as receiving data from client computing objects or devices 2320, 2322, 2324, 2326, 2328, *etc.*, storing of data, processing of data, transmitting data to client computing objects or devices 2320, 2322, 2324, 2326, 2328, *etc.*, although any computer can be considered a client, a server, or both, depending on the circumstances. Any of these computing devices may be processing data, or requesting transaction services or tasks that may implicate the techniques for systems as described herein for one or more embodiments.

**[00163]** A server is typically a remote computer system accessible over a remote or local network, such as the Internet or wireless network infrastructures. The client process may be active in a first computer system, and the server process may be active in a second computer system, communicating with one another over a communications medium, thus providing distributed functionality and allowing multiple clients to take advantage of the information-gathering capabilities of the server. Any software objects utilized pursuant to the techniques described herein can be provided standalone, or distributed across multiple computing devices or objects.

**[00164]** In a network environment in which the communications network/bus 2340 is the Internet, for example, the computing objects 2310,



LSTP102WO

2312, *etc.* can be Web servers, file servers, media servers, *etc.* with which the client computing objects or devices 2320, 2322, 2324, 2326, 2328, *etc.* communicate via any of a number of known protocols, such as the hypertext transfer protocol (HTTP). Computing objects 2310, 2312, *etc.* may also serve as client computing objects or devices 2320, 2322, 2324, 2326, 2328, *etc.*, as may be characteristic of a distributed computing environment.

#### EXEMPLARY COMPUTING DEVICE

**[00165]** As mentioned, advantageously, the techniques described herein can be applied to any suitable device. It is to be understood, therefore, that handheld, portable and other computing devices and computing objects of all kinds are contemplated for use in connection with the various embodiments. Accordingly, the below computer described below in FIG. 24 is but one example of a computing device. Additionally, a suitable server can include one or more aspects of the below computer, such as a media server or other media management server components.

**[00166]** Although not required, embodiments can partly be implemented via an operating system, for use by a developer of services for a device or object, and/or included within application software that operates to perform one or more functional aspects of the various embodiments described herein. Software may be described in the general context of computer executable instructions, such as program modules, being executed by one or more computers, such as client workstations, servers or other devices. Those skilled in the art will appreciate that computer systems have a variety of configurations and protocols that can be used to communicate data, and thus, no particular configuration or protocol is to be considered limiting.

**[00167]** FIG. 24 thus illustrates an example of a suitable computing system environment 2400 in which one or aspects of the embodiments described herein can be implemented, although as made clear above, the computing system environment 2400 is only one example of a suitable computing environment and is not intended to suggest any limitation as to scope of use or functionality. Neither is the computing system environment 2400 be interpreted as having any dependency or requirement relating to any

LSTP102WO

one or combination of components illustrated in the exemplary computing system environment 2400.

**[00168]** With reference to FIG. 24, an exemplary computing device for implementing one or more embodiments in the form of a computer 2410 is depicted. Components of computer 2410 may include, but are not limited to, a processing unit 2420, a system memory 2430, and a system bus 2422 that couples various system components including the system memory to the processing unit 2420. Processing unit 2420 may, for example, perform functions associated with processor(s) 214 of inertia estimator 202, while system memory 2430 may perform functions associated with memory 216.

**[00169]** Computer 2410 typically includes a variety of computer readable media and can be any available media that can be accessed by computer 2410. The system memory 2430 may include computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) and/or random access memory (RAM). By way of example, and not limitation, system memory 2430 may also include an operating system, application programs, other program modules, and program data.

**[00170]** A user can enter commands and information into the computer 2410 through input devices 2440, non-limiting examples of which can include a keyboard, keypad, a pointing device, a mouse, stylus, touchpad, touchscreen, trackball, motion detector, camera, microphone, joystick, game pad, scanner, or any other device that allows the user to interact with computer 2410. A monitor or other type of display device is also connected to the system bus 2422 via an interface, such as output interface 2450. In addition to a monitor, computers can also include other peripheral output devices such as speakers and a printer, which may be connected through output interface 2450. In one or more embodiments, input devices 2440 can provide user input to interface component 212, while output interface 2450 can receive information relating to operations of inertia estimator 202 from interface component 212.

**[00171]** The computer 2410 may operate in a networked or distributed environment using logical connections to one or more other remote computers, such as remote computer 2470. The remote computer 2470 may be a personal computer, a server, a router, a network PC, a peer device or

LSTP102WO

other common network node, or any other remote media consumption or transmission device, and may include any or all of the elements described above relative to the computer 2410. The logical connections depicted in FIG. 24 include a network 2472, such local area network (LAN) or a wide area network (WAN), but may also include other networks/buses *e.g.*, cellular networks.

**[00172]** As mentioned above, while exemplary embodiments have been described in connection with various computing devices and network architectures, the underlying concepts may be applied to any network system and any computing device or system in which it is desirable to publish or consume media in a flexible way.

**[00173]** Also, there are multiple ways to implement the same or similar functionality, *e.g.*, an appropriate API, tool kit, driver code, operating system, control, standalone or downloadable software object, *etc.* which enables applications and services to take advantage of the techniques described herein. Thus, embodiments herein are contemplated from the standpoint of an API (or other software object), as well as from a software or hardware object that implements one or more aspects described herein. Thus, various embodiments described herein can have aspects that are wholly in hardware, partly in hardware and partly in software, as well as in software.

**[00174]** The word "exemplary" is used herein to mean serving as an example, instance, or illustration. For the avoidance of doubt, the aspects disclosed herein are not limited by such examples. In addition, any aspect or design described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other aspects or designs, nor is it meant to preclude equivalent exemplary structures and techniques known to those of ordinary skill in the art. Furthermore, to the extent that the terms "includes," "has," "contains," and other similar words are used in either the detailed description or the claims, for the avoidance of doubt, such terms are intended to be inclusive in a manner similar to the term "comprising" as an open transition word without precluding any additional or other elements.

**[00175]** Computing devices typically include a variety of media, which can include computer-readable storage media (*e.g.*, memory 216 or 1112) and/or communications media, in which these two terms are used herein



## LSTP102WO

differently from one another as follows. Computer-readable storage media can be any available storage media that can be accessed by the computer, is typically of a non-transitory nature, and can include both volatile and nonvolatile media, removable and non-removable media. By way of example, and not limitation, computer-readable storage media can be implemented in connection with any method or technology for storage of information such as computer-readable instructions, program modules, structured data, or unstructured data. Computer-readable storage media can include, but are not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disk (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or other tangible and/or non-transitory media which can be used to store desired information. Computer-readable storage media can be accessed by one or more local or remote computing devices, *e.g.*, via access requests, queries or other data retrieval protocols, for a variety of operations with respect to the information stored by the medium.

**[00176]** On the other hand, communications media typically embody computer-readable instructions, data structures, program modules or other structured or unstructured data in a data signal such as a modulated data signal, *e.g.*, a carrier wave or other transport mechanism, and includes any information delivery or transport media. The term “modulated data signal” or signals refers to a signal that has one or more of its characteristics set or changed in such a manner as to encode information in one or more signals. By way of example, and not limitation, communication media include wired media, such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media.

**[00177]** As mentioned, the various techniques described herein may be implemented in connection with hardware or software or, where appropriate, with a combination of both. As used herein, the terms “component,” “system” and the like are likewise intended to refer to a computer-related entity, either hardware, a combination of hardware and software, software, or software in execution. For example, a component may be, but is not limited to being, a process running on a processor, a processor, an object, an executable, a thread of execution, a program, and/or a computer. By way of illustration,

LSTP102WO

both an application running on computer and the computer can be a component. One or more components may reside within a process and/or thread of execution and a component may be localized on one computer and/or distributed between two or more computers. Further, a "device" can come in the form of specially designed hardware; generalized hardware made specialized by the execution of software thereon that enables the hardware to perform specific function (*e.g.*, coding and/or decoding); software stored on a computer readable medium; or a combination thereof.

**[00178]** The aforementioned systems have been described with respect to interaction between several components. It can be appreciated that such systems and components can include those components or specified sub-components, some of the specified components or sub-components, and/or additional components, and according to various permutations and combinations of the foregoing. Sub-components can also be implemented as components communicatively coupled to other components rather than included within parent components (hierarchical). Additionally, it is to be noted that one or more components may be combined into a single component providing aggregate functionality or divided into several separate sub-components, and that any one or more middle layers, such as a management layer, may be provided to communicatively couple to such sub-components in order to provide integrated functionality. Any components described herein may also interact with one or more other components not specifically described herein but generally known by those of skill in the art.

**[00179]** In order to provide for or aid in the numerous inferences described herein (*e.g.* inferring audio segments), components described herein can examine the entirety or a subset of the data to which it is granted access and can provide for reasoning about or infer states of the system, environment, *etc.* from a set of observations as captured *via* events and/or data. Inference can be employed to identify a specific context or action, or can generate a probability distribution over states, for example. The inference can be probabilistic - that is, the computation of a probability distribution over states of interest based on a consideration of data and events. Inference can also refer to techniques employed for composing higher-level events from a set of events and/or data.

LSTP102WO

**[00180]** Such inference can result in the construction of new events or actions from a set of observed events and/or stored event data, whether or not the events are correlated in close temporal proximity, and whether the events and data come from one or several event and data sources. Various classification (explicitly and/or implicitly trained) schemes and/or systems (e.g., support vector machines, neural networks, expert systems, Bayesian belief networks, fuzzy logic, data fusion engines, *etc.*) can be employed in connection with performing automatic and/or inferred action in connection with the claimed subject matter.

**[00181]** A classifier can map an input attribute vector,  $x = (x_1, x_2, x_3, x_4, \dots, x_n)$ , to a confidence that the input belongs to a class, as by  $f(x) = \text{confidence}(\text{class})$ . Such classification can employ a probabilistic and/or statistical-based analysis (e.g., factoring into the analysis utilities and costs) to prognose or infer an action that a user desires to be automatically performed. A support vector machine (SVM) is an example of a classifier that can be employed. The SVM operates by finding a hyper-surface in the space of possible inputs, where the hyper-surface attempts to split the triggering criteria from the non-triggering events. Intuitively, this makes the classification correct for testing data that is near, but not identical to training data. Other directed and undirected model classification approaches include, e.g., naïve Bayes, Bayesian networks, decision trees, neural networks, fuzzy logic models, and probabilistic classification models providing different patterns of independence can be employed. Classification as used herein also is inclusive of statistical regression that is utilized to develop models of priority.

**[00182]** In view of the exemplary systems described above, methodologies that may be implemented in accordance with the described subject matter will be better appreciated with reference to the flowcharts of the various figures (e.g., FIGs. 9, 10, and 20-22). While for purposes of simplicity of explanation, the methodologies are shown and described as a series of blocks, it is to be understood and appreciated that the claimed subject matter is not limited by the order of the blocks, as some blocks may occur in different orders and/or concurrently with other blocks from what is depicted and described herein. Where non-sequential, or branched, flow is illustrated via flowchart, it can be appreciated that various other branches, flow paths, and

LSTP102WO

orders of the blocks, may be implemented which achieve the same or a similar result. Moreover, not all illustrated blocks may be required to implement the methodologies described hereinafter.

**[00183]** In addition to the various embodiments described herein, it is to be understood that other similar embodiments can be used or modifications and additions can be made to the described embodiment(s) for performing the same or equivalent function of the corresponding embodiment(s) without deviating there from. Still further, multiple processing chips or multiple devices can share the performance of one or more functions described herein, and similarly, storage can be effected across a plurality of devices. Accordingly, the invention is not to be limited to any single embodiment, but rather can be construed in breadth, spirit and scope in accordance with the appended claims.

LSTP102WO

## CLAIMS

What is claimed is:

1. A method for estimating parameters of a motion system, comprising:  
generating a torque command signal that varies continuously over time;  
measuring velocity data for a motion device representing a velocity of the motion system in response to the torque command signal; and  
determining at least one of an inertia or a friction coefficient of the motion system based at least in part on the velocity data and the torque command signal.
2. The method of claim 1, wherein the determining comprises determining the at least one of the inertia or the friction coefficient based at least in part on one or more integrals of the velocity data over a time range and one or more integrals of the torque command signal over the time range.
3. The method of claim 1, wherein the generating the torque command signal comprises adjusting the torque command signal in accordance with a predefined testing sequence, wherein the adjusting includes changing at least one of a direction or a rate of change of the torque command signal in response to the velocity of the motion system reaching a predefined velocity checkpoint.
4. The method of claim 2, wherein the determining comprises:  
designating a first segment of the time range as an acceleration phase;  
designating a second segment of the time range as a deceleration phase;  
integrating the torque command signal and the velocity data over the acceleration phase to yield  $U_{acc}$  and  $V_{acc}$ , respectively;  
integrating the torque command signal and the velocity data over the deceleration phase to yield  $U_{dec}$  and  $V_{dec}$ , respectively; and  
determining the at least one of the inertia or the friction coefficient as a function of  $U_{acc}$ ,  $V_{acc}$ ,  $U_{dec}$ , and  $V_{dec}$ .

LSTP102WO

where:

$$U_{acc} = \int u_{acc}(t),$$

$$V_{acc} = \int v_{acc}(t),$$

$$U_{dec} = \int u_{dec}(t),$$

$$V_{dec} = \int v_{dec}(t),$$

$u_{acc}(t)$  is a portion of the torque command signal corresponding to the acceleration phase,

$v_{acc}(t)$  is a portion of the velocity data corresponding to the acceleration phase,

$u_{dec}(t)$  is a portion of the torque command signal corresponding to the deceleration phase, and

$v_{dec}(t)$  is a portion of the velocity data corresponding to the deceleration phase.

5. The method of claim 4, wherein the determining the at least one of the inertia or the friction coefficient comprises at least one of determining the inertia according to:

$$J = \frac{U_{dec}V_{acc} - U_{acc}V_{dec}}{\Delta v_{dec}(t)V_{acc} - \Delta v_{acc}(t)V_{dec}},$$

or determining the friction coefficient according to:

$$B = \frac{\Delta v_{dec}(t)U_{acc} - \Delta v_{acc}(t)U_{dec}}{\Delta v_{dec}(t)V_{acc} - \Delta v_{acc}(t)V_{dec}},$$

where:

$J$  is the inertia,

$B$  is the friction coefficient,

$\Delta v_{acc}(t)$  is a difference between a velocity of the motion system at an

LSTP102WO

end of the acceleration phase and a velocity of the motion system at a beginning of the acceleration phase, and

$\Delta v_{dec}(t)$  is a difference between a velocity of the motion system at an end of the deceleration phase and a velocity of the motion system at a beginning of the deceleration phase.

6. The method of claim 1, further comprising determining at least one controller gain coefficient for the motion system based on the at least one of the inertia or the friction coefficient.
7. A system for estimating mechanical parameters of a motion system, comprising:
  - a memory;
  - a processor configured to execute computer-executable components stored on the memory, the computer-executable components comprising:
    - a torque command generator configured to generate a torque command signal that varies continuously over time during a testing sequence;
    - a velocity monitoring component configured to obtain velocity data representing a velocity of a motion system over time in response to the torque command signal; and
    - at least one of:
      - an inertia component configured to estimate an inertia of the motion system based on the torque command signal and the velocity data, or
      - a friction coefficient component configured to estimate a friction coefficient of the motion system based on the torque command signal and the velocity data.
8. The system of claim 7, wherein the at least one of the inertia component or the friction coefficient component is further configured to estimate the inertia or the friction coefficient, respectively, based at least in part on one or more integrals of the velocity data over a time range and one or more integrals of the torque command signal over the time range, wherein the



LSTP102WO

time range is at least a portion of a duration of the testing sequence.

9. The system of claim 7, wherein the torque command generator is further configured to control the torque command signal in accordance with a torque function  $u(t)$ , where  $u(t)$  is based on a set of predefined instructions associated with respective phases of the testing sequence, and wherein the respective phases are triggered in response to the velocity of the motion system reaching respective defined velocity checkpoint values.

10. The system of claim 8, wherein the at least one of the inertia component or the friction coefficient component is further configured to estimate the inertia as a function of  $U_{acc}$ ,  $V_{acc}$ ,  $U_{dec}$ , and  $V_{dec}$ ,

where:

$$U_{acc} = \int u_{acc}(t),$$

$$V_{acc} = \int v_{acc}(t),$$

$$U_{dec} = \int u_{dec}(t),$$

$$V_{dec} = \int v_{dec}(t),$$

$u_{acc}(t)$  is a portion of the torque command signal corresponding to an acceleration phase of the testing sequence,

$v_{acc}(t)$  is a portion of the velocity data corresponding to the acceleration phase,

$u_{dec}(t)$  is a portion of the torque command signal corresponding to a deceleration phase of the testing sequence, and

$v_{dec}(t)$  is a portion of the velocity data corresponding to the deceleration phase.

11. The system of claim 10, wherein the inertia component is further configured to estimate the inertia based on:

$$J = \frac{U_{dec}V_{acc} - U_{acc}V_{dec}}{\Delta v_{dec}(t)V_{acc} - \Delta v_{acc}(t)V_{dec}},$$



LSTP102WO

where:

$J$  is the inertia,

$\Delta v_{acc}(t)$  is a difference between a velocity of the motion system at an end of the acceleration phase and a velocity of the motion system at a beginning of the acceleration phase, and

$\Delta v_{dec}(t)$  is a difference between a velocity of the motion system at an end of the deceleration phase and a velocity of the motion system at a beginning of the deceleration phase.

12. The system of claim 10, wherein the friction coefficient component is further configured to estimate the friction coefficient based on:

$$B = \frac{\Delta v_{dec}(t)U_{acc} - \Delta v_{acc}U_{dec}}{\Delta v_{dec}(t)V_{acc} - \Delta v_{acc}(t)V_{dec}},$$

where,

$B$  is the friction coefficient,

$\Delta v_{acc}(t)$  is a difference between a velocity of the motion system at an end of the acceleration phase and a velocity of the motion system at a beginning of the acceleration phase, and

$\Delta v_{dec}(t)$  is a difference between a velocity of the motion system at an end of the deceleration phase and a velocity of the motion system at a beginning of the deceleration phase.

13. The system of claim 7, further comprising a tuning component configured to generate at least one controller gain coefficient as a function of at least one of the inertia or the friction coefficient.

14. A method for generating a motion profile, comprising:  
receiving a setpoint indicating at least one of a target position or a target velocity for a motion device; and  
generating a motion profile for transitioning the motion device to the at

LSTP102WO

least one of the target position or the target velocity, the motion profile defining a jerk reference that varies continuously as a function of time for at least one segment of the motion profile.

15. The method of claim 14, wherein the generating the motion profile includes defining, as a function of time, at least one of an acceleration reference, a velocity reference, or a position reference for the at least one segment of the motion profile.
16. The method of claim 14, wherein the generating comprises:  
calculating at least one of the jerk reference, an acceleration reference, a velocity reference, or a position reference for respective segments of the motion profile;  
calculating time durations for the respective segments of the motion profile;  
rounding the time durations to respective nearest multiples of a sample time of a motion controller to yield rounded time durations; and  
recalculating at least one of the jerk reference, the acceleration reference, the deceleration reference, or the velocity reference using the rounded time durations.
17. The method of claim 14, wherein the generating comprises generating the motion profile according to the relationships:

$$t_1^3(t_1 + t_2) = t_4^3(t_4 + t_5) ,$$

$$t_1 = \frac{3A}{2J},$$

$$t_4 = \frac{3D}{2I},$$

$$t_2 = \frac{V}{A} - \frac{3A}{2J},$$

LSTP102WO

$$t_5 = \frac{V}{D} - \frac{3D}{2I},$$

$$t_3 = \frac{P}{V} - \frac{3A}{4J} - \frac{V}{2A} - \frac{3D}{4I} - \frac{V}{2D},$$

$$\frac{A^2}{D^2} = \frac{J^3}{I^3},$$

$$V \geq \frac{3A^2}{2J} = \frac{3}{2I\rho^{\frac{4}{3}}} D^2,$$

$$V \geq \frac{3}{2I} D^2,$$

$$\frac{P}{V} \geq \frac{3}{4I} \rho^{-\frac{1}{3}} D + \frac{\rho V}{2D} + \frac{3}{4I} D + \frac{1}{2} \frac{V}{D},$$

$$\frac{P}{V} \geq \left( \rho^{-\frac{1}{3}} + 1 \right) \frac{3}{4I} D + \frac{1}{2} (\rho + 1) \frac{V}{D}, \text{ and}$$

$$\frac{I}{J} = \rho^{\frac{2}{3}}$$

where:

$P$  is a position of the motion device,

$J$  is a maximum acceleration jerk,

$I$  is a maximum deceleration jerk,

$A$  is a maximum acceleration,

$D$  is a maximum deceleration,

$V$  is a maximum velocity,

$t_1$  is a duration of an increasing acceleration stage and a decreasing acceleration stage of the motion profile,

$t_2$  is a duration of a constant acceleration stage of the motion profile,

$t_3$  is a duration of a constant velocity stage of the motion profile,

$t_4$  is a duration of an increasing deceleration stage and a decreasing

LSTP102WO

deceleration stage of the motion profile,

$t_5$  is a duration of a constant deceleration stage of the motion profile,

and

$$\rho = \frac{D}{A}.$$

18. A system for generating a motion profile, comprising:  
a memory;  
a processor configured to execute computer-executable components stored on the memory, the computer-executable components comprising:  
a motion profile generator configured to generate, in response to receipt of a target position or a target velocity for a motion device, a motion profile having a continuous jerk reference that varies continuously over time for at least one stage of the motion profile, wherein the motion profile defines a trajectory for transitioning the motion device to the target position or the target velocity.
19. The system of claim 18, wherein the motion profile generator is further configured to generate the motion profile as a function of at least one defined constraint, wherein the at least one defined constraint includes at least one of a sample time, a velocity limit, an acceleration limit, a jerk limit, or a deceleration limit.
20. The system of claim 18, wherein the motion profile generator is further configured to generate the motion profile according to the relationships:

$$t_1^3(t_1 + t_2) = t_4^3(t_4 + t_5),$$

$$t_1 = \frac{3A}{2J},$$

$$t_4 = \frac{3D}{2I},$$

LSTP102WO

$$t_2 = \frac{V}{A} - \frac{3A}{2J},$$

$$t_5 = \frac{V}{D} - \frac{3D}{2I},$$

$$t_3 = \frac{P}{V} - \frac{3A}{4J} - \frac{V}{2A} - \frac{3D}{4I} - \frac{V}{2D},$$

$$\frac{A^2}{D^2} = \frac{J^3}{I^3},$$

$$V \geq \frac{3A^2}{2J} = \frac{3}{2I\rho^{\frac{4}{3}}} D^2,$$

$$V \geq \frac{3}{2I} D^2,$$

$$\frac{P}{V} \geq \frac{3}{4I} \rho^{-\frac{1}{3}} D + \frac{\rho V}{2D} + \frac{3}{4I} D + \frac{1}{2} \frac{V}{D},$$

$$\frac{P}{V} \geq \left( \rho^{-\frac{1}{3}} + 1 \right) \frac{3}{4I} D + \frac{1}{2} (\rho + 1) \frac{V}{D}, \text{ and}$$

$$\frac{I}{J} = \rho^{\frac{2}{3}}$$

where:

$P$  is a position of the motion device,

$J$  is a maximum acceleration jerk,

$I$  is a maximum deceleration jerk,

$A$  is a maximum acceleration,

$D$  is a maximum deceleration,

$V$  is a maximum velocity,

$t_1$  is a duration of an increasing acceleration stage and a decreasing acceleration stage of the motion profile,

LSTP102WO

$t_2$  is a duration of a constant acceleration stage of the motion profile,

$t_3$  is a duration of a constant velocity stage of the motion profile,

$t_4$  is a duration of an increasing deceleration stage and a decreasing deceleration stage of the motion profile,

$t_5$  is a duration of a constant deceleration stage of the motion profile,

and

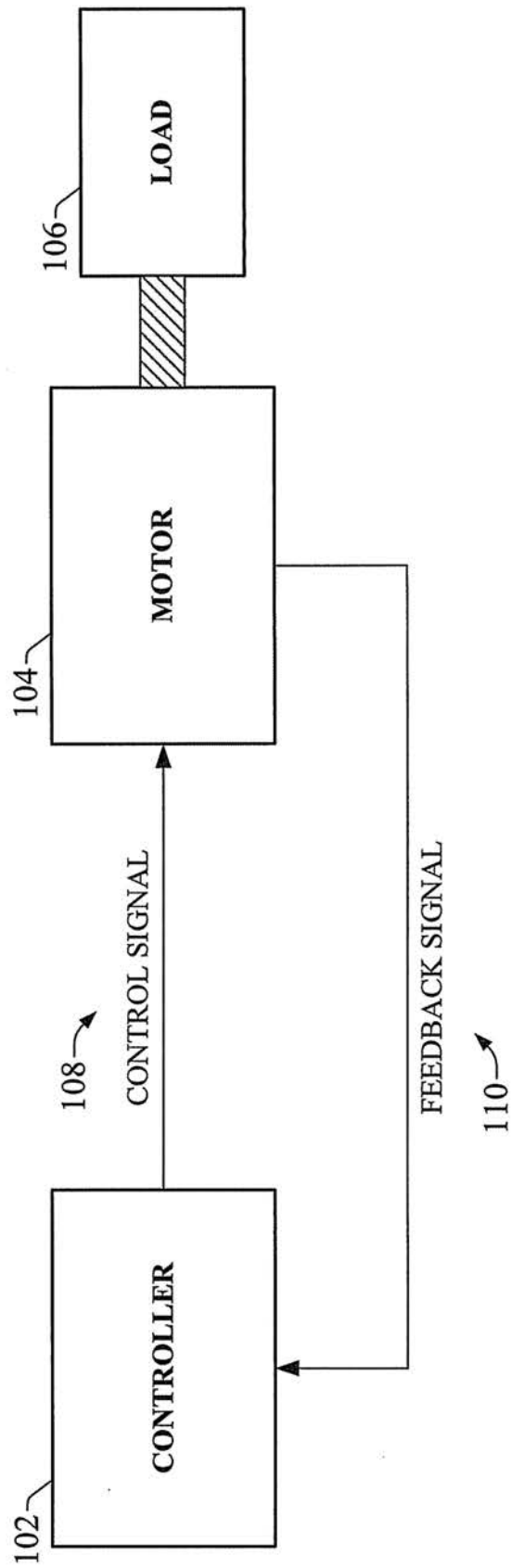
$$\rho = \frac{D}{A}.$$

LSTP102WO

## ABSTRACT

Systems and methods for estimating an inertia and a friction coefficient for a controlled mechanical system are provided. In one or more embodiments, an inertia estimator can generate a torque command signal that varies continuously over time during a testing sequence. The velocity of a motion system in response to the time-varying torque command signal is measured and recorded during the testing sequence. The inertia estimator then estimates the inertia and/or the friction coefficient of the motion system based on the torque command data sent to the motion system and the measured velocity data. Systems and methods are also provided for generating a constraint-based, time-optimal motion profile for controlling the trajectory of a point-to-point move in a motion control system. In one or more embodiments, a profile generator can calculate an ST-curve motion profile that includes a jerk reference that varies continuously over time for at least one of the motion profile segments, thereby producing a smooth, time-optimal trajectory.

1/22

**FIG. 1**



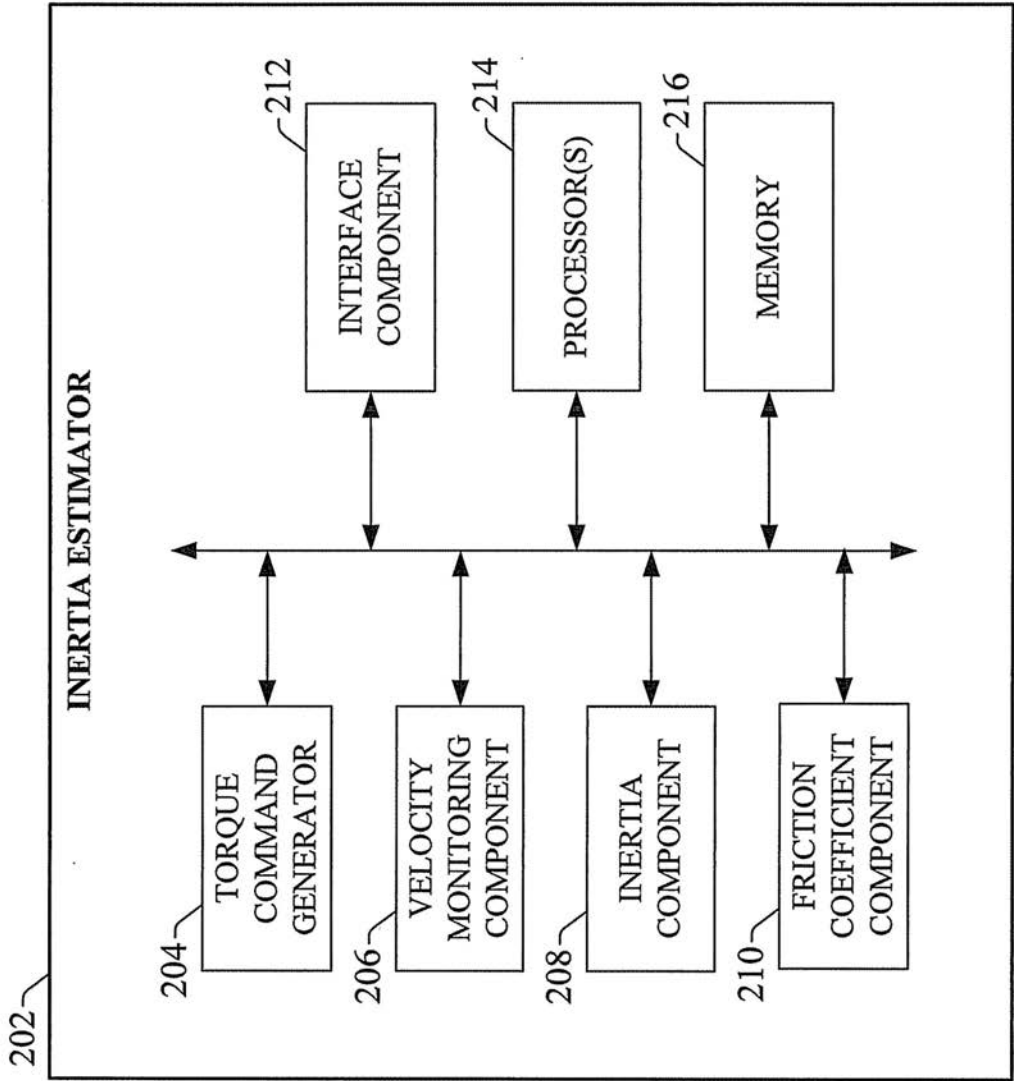
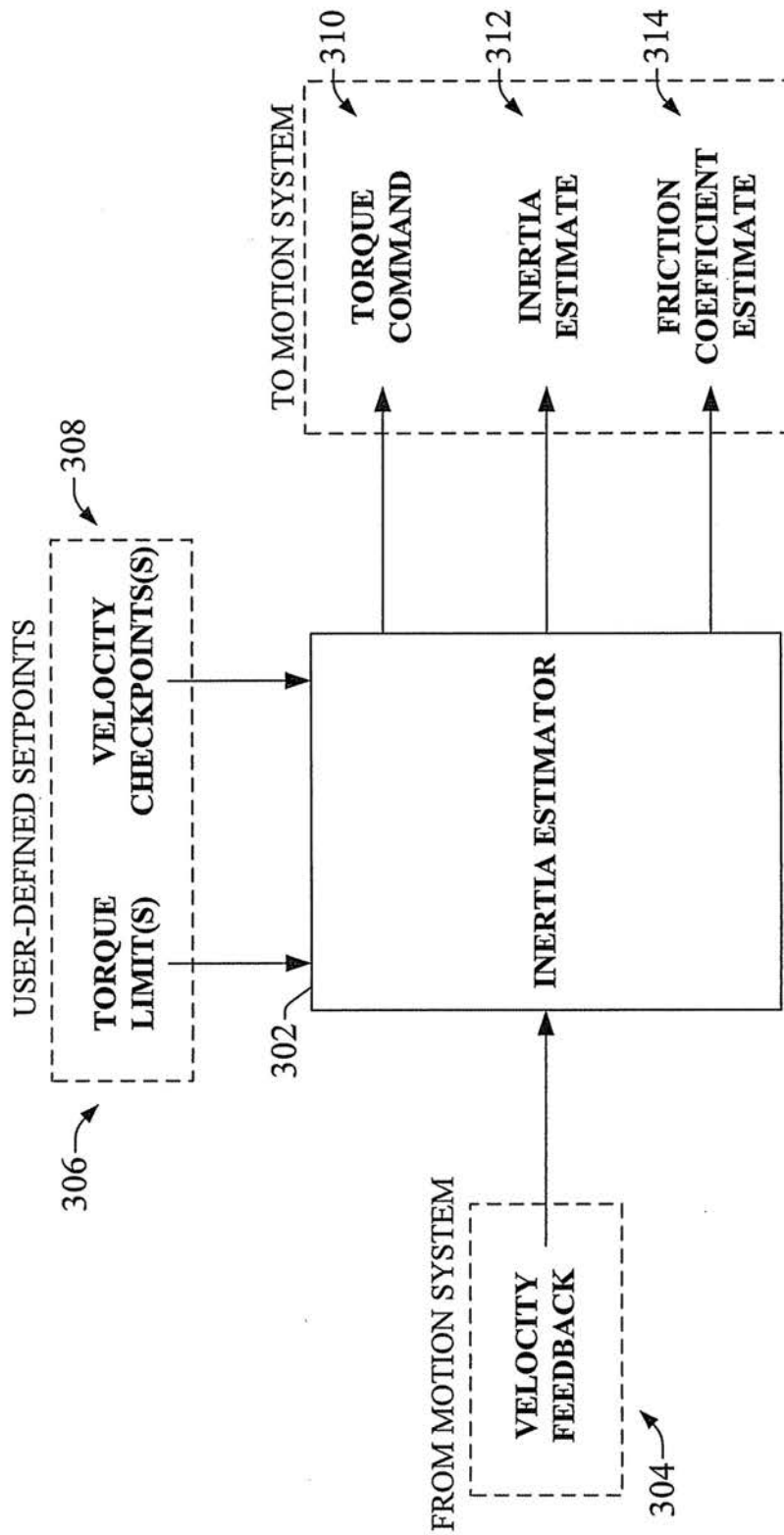


FIG. 2

3/22

**FIG. 3**

4/22

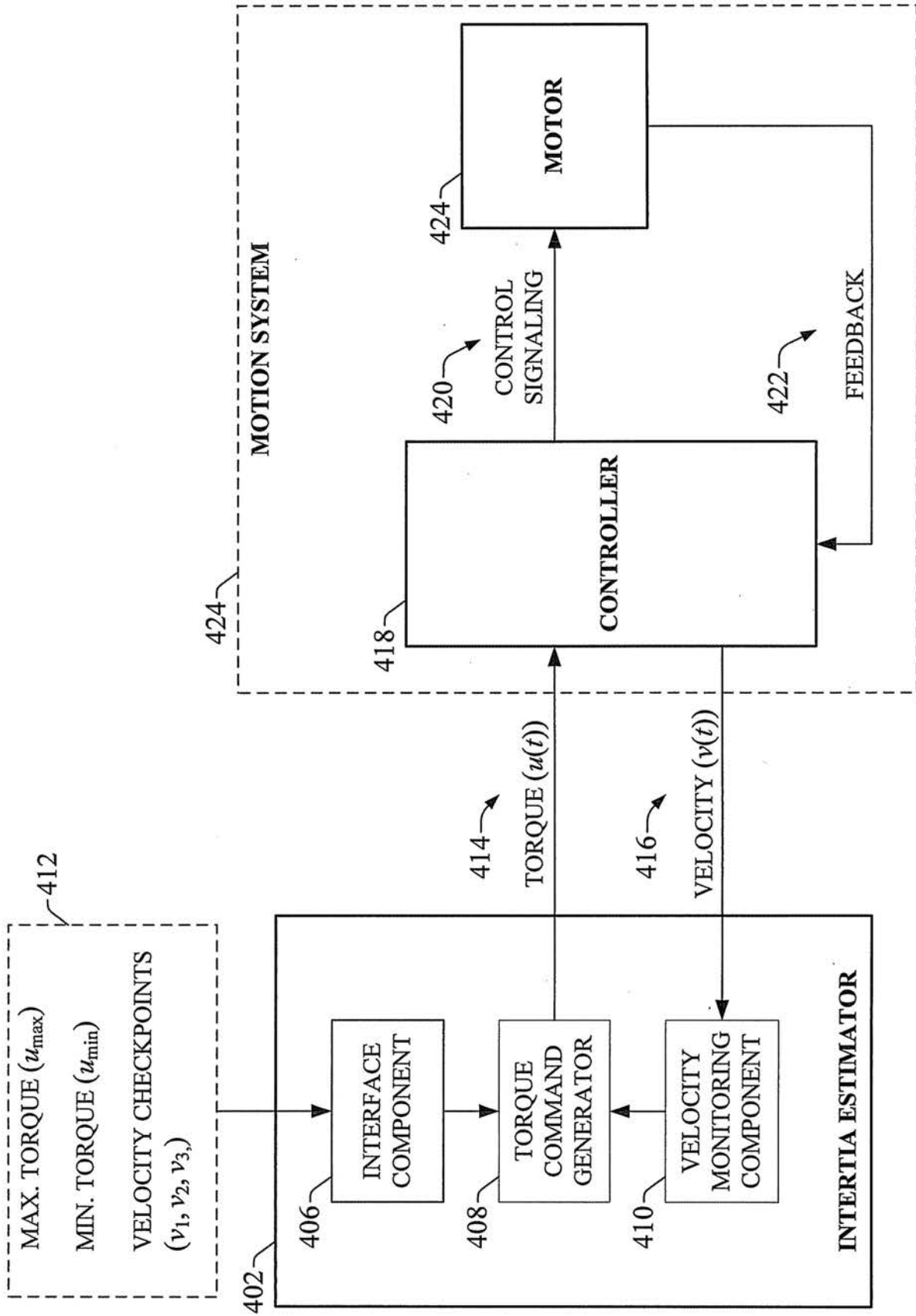


FIG. 4

5/22

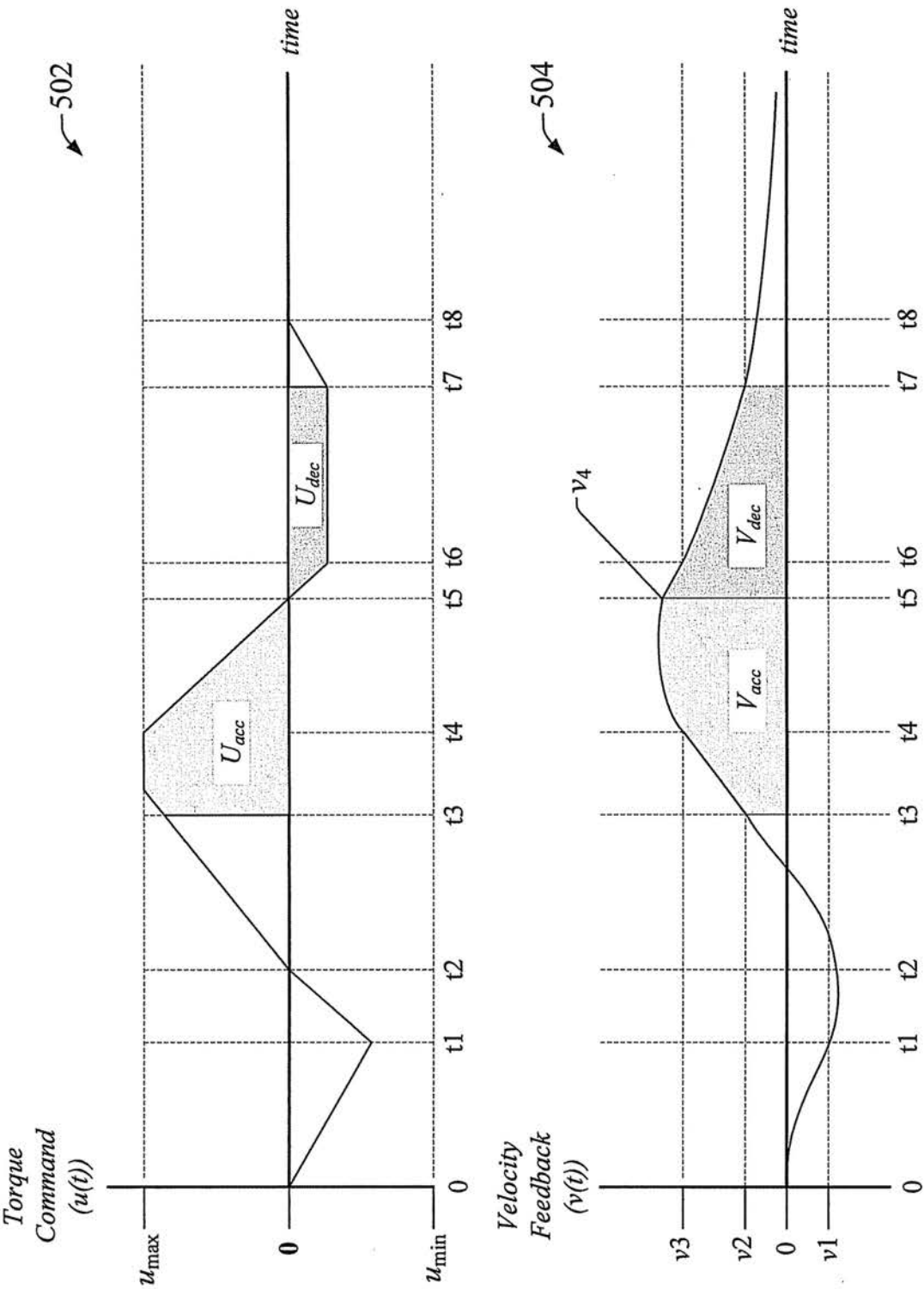


FIG. 5

6/22

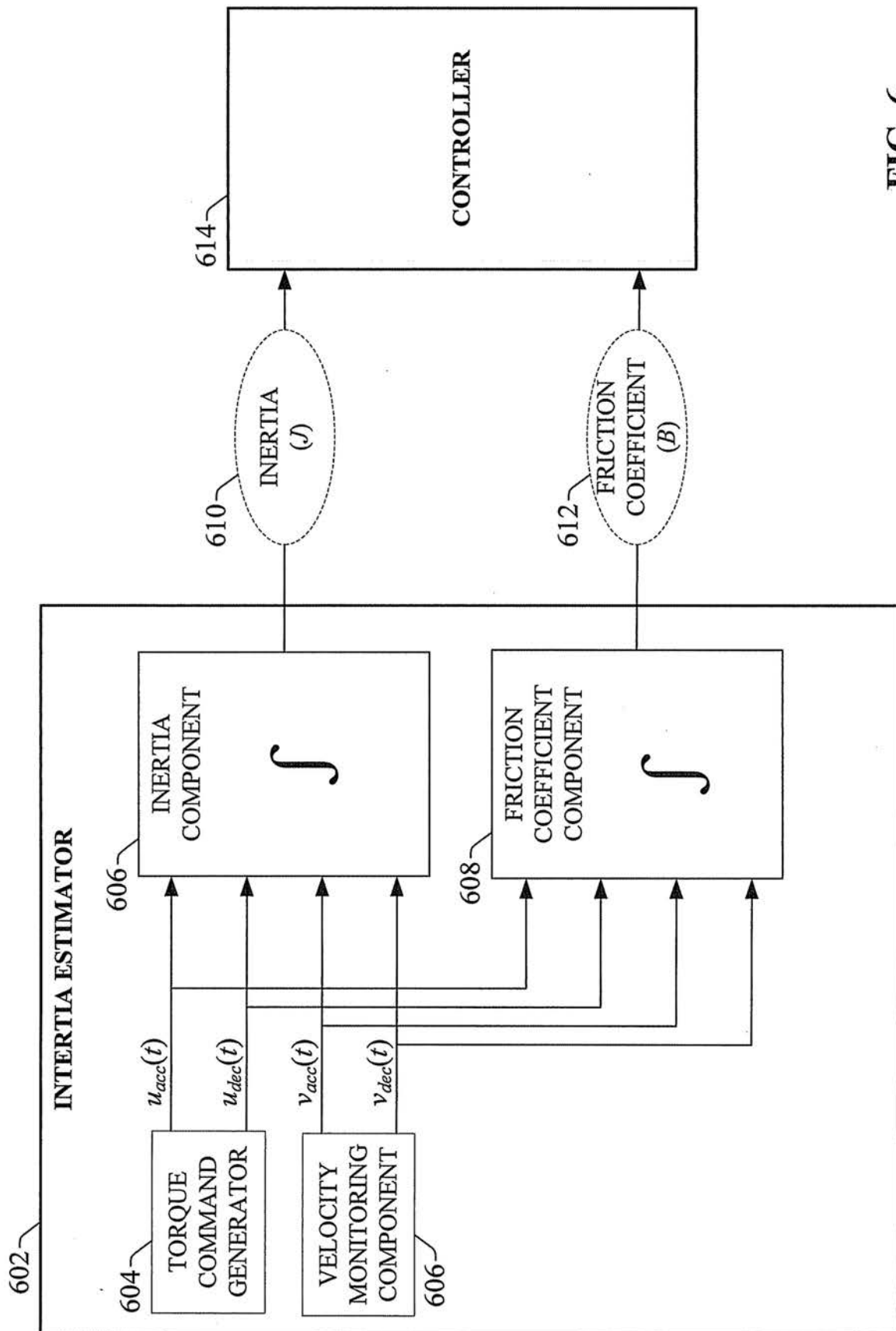
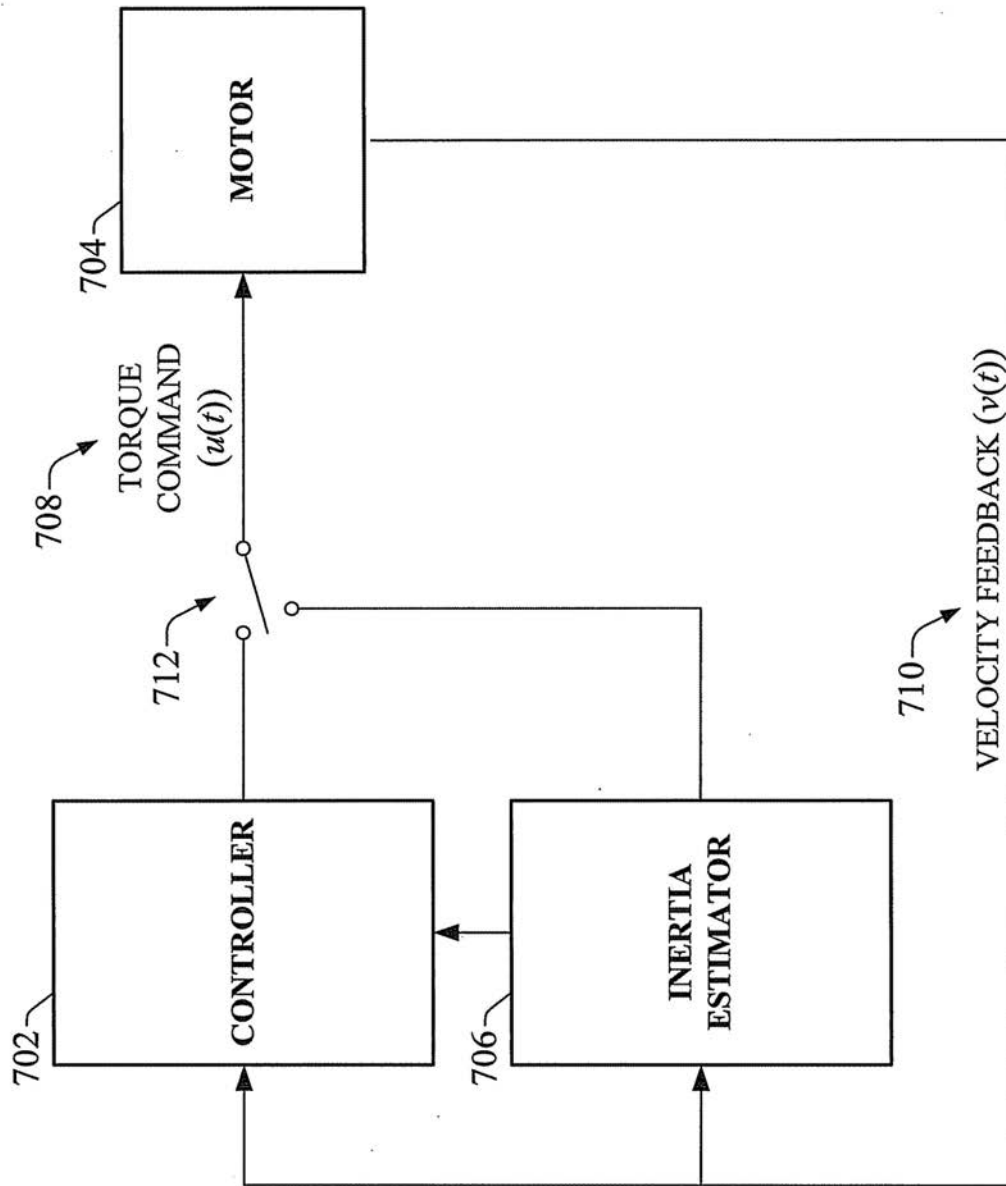
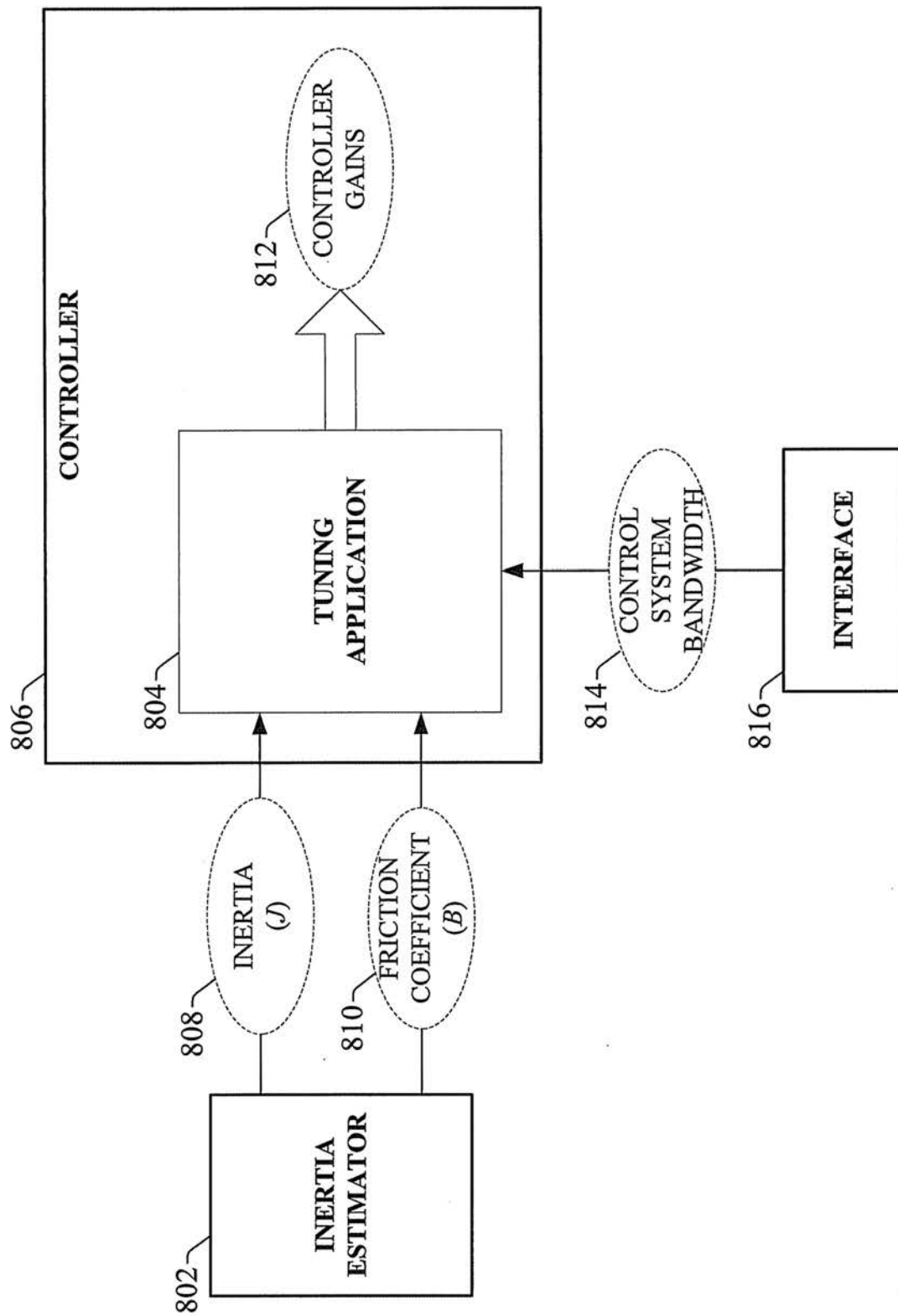


FIG. 6

7/22

**FIG. 7**

8/22

**FIG. 8**

9/22

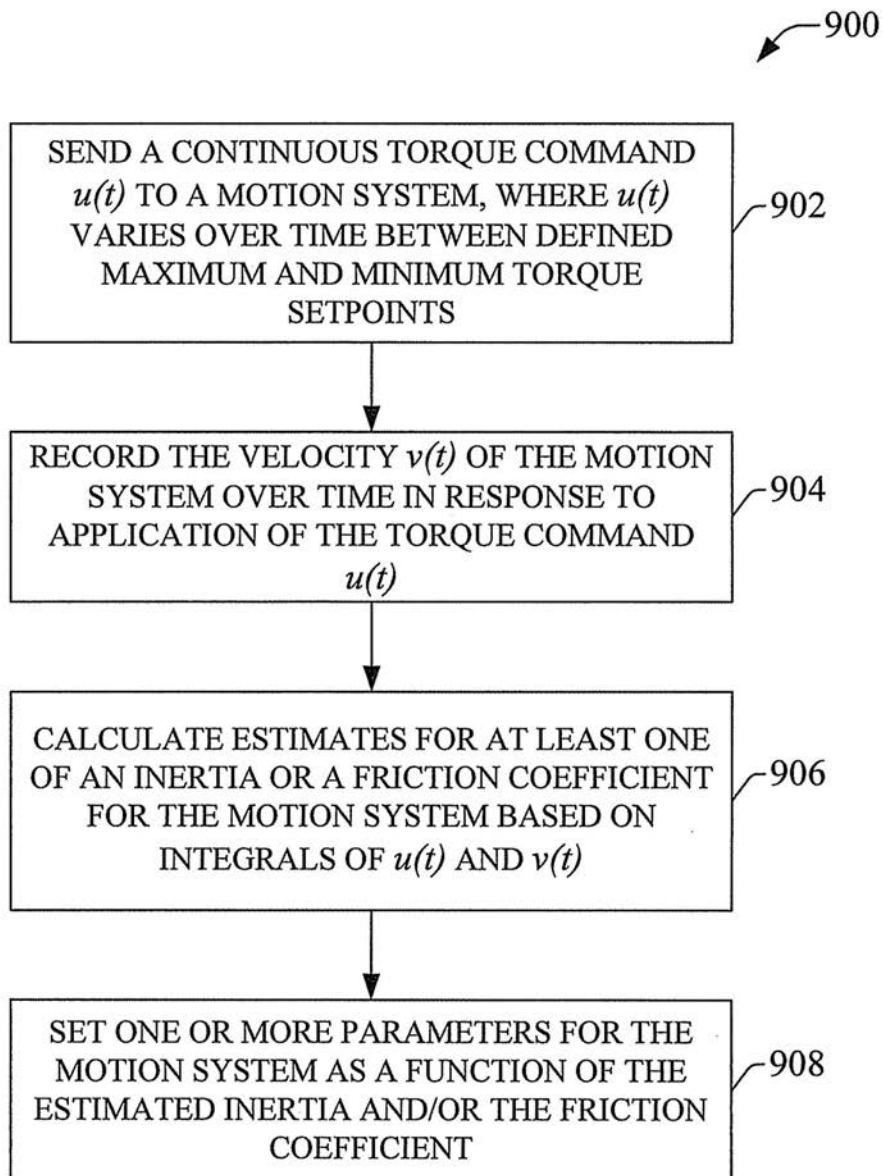


FIG. 9



10/22

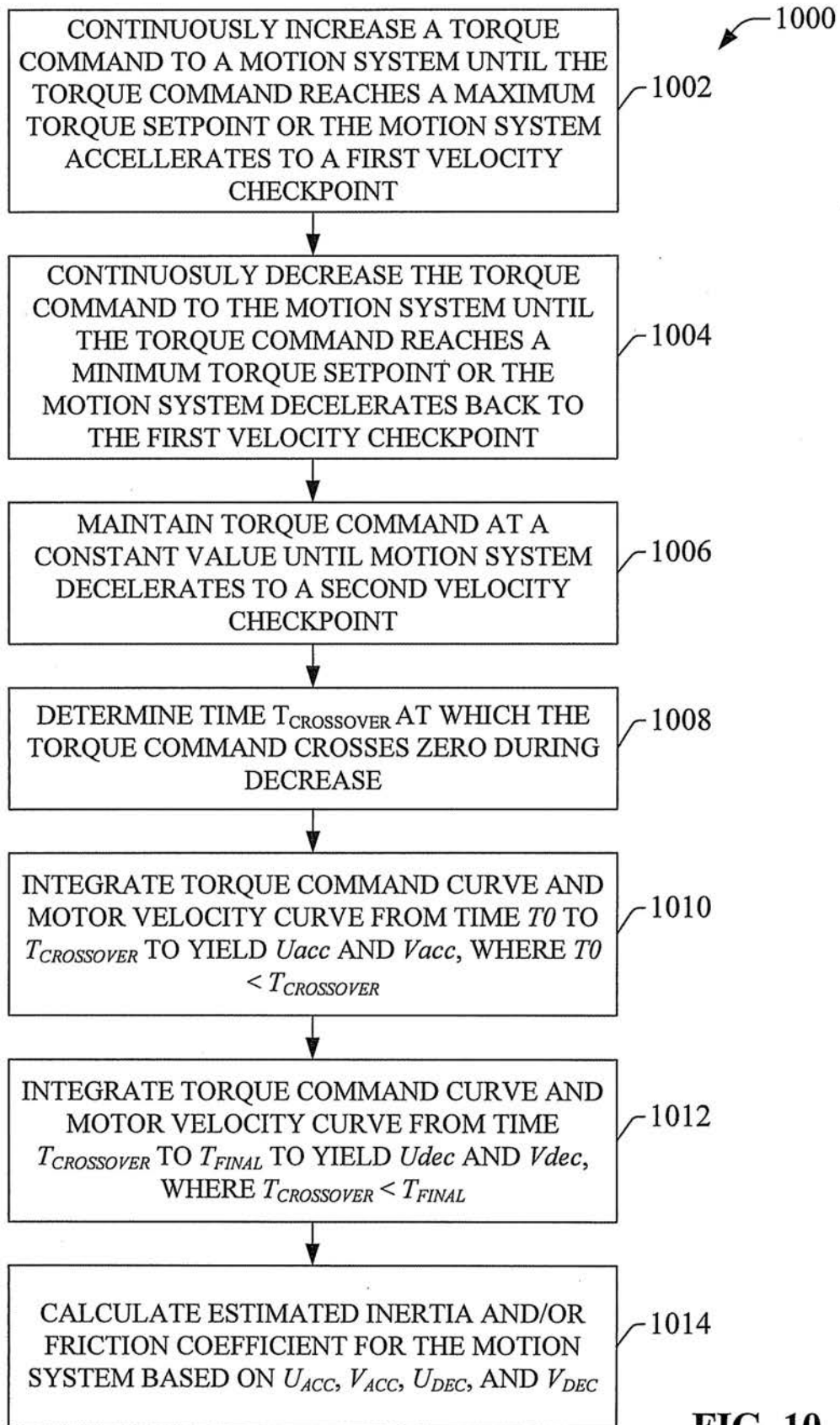


FIG. 10

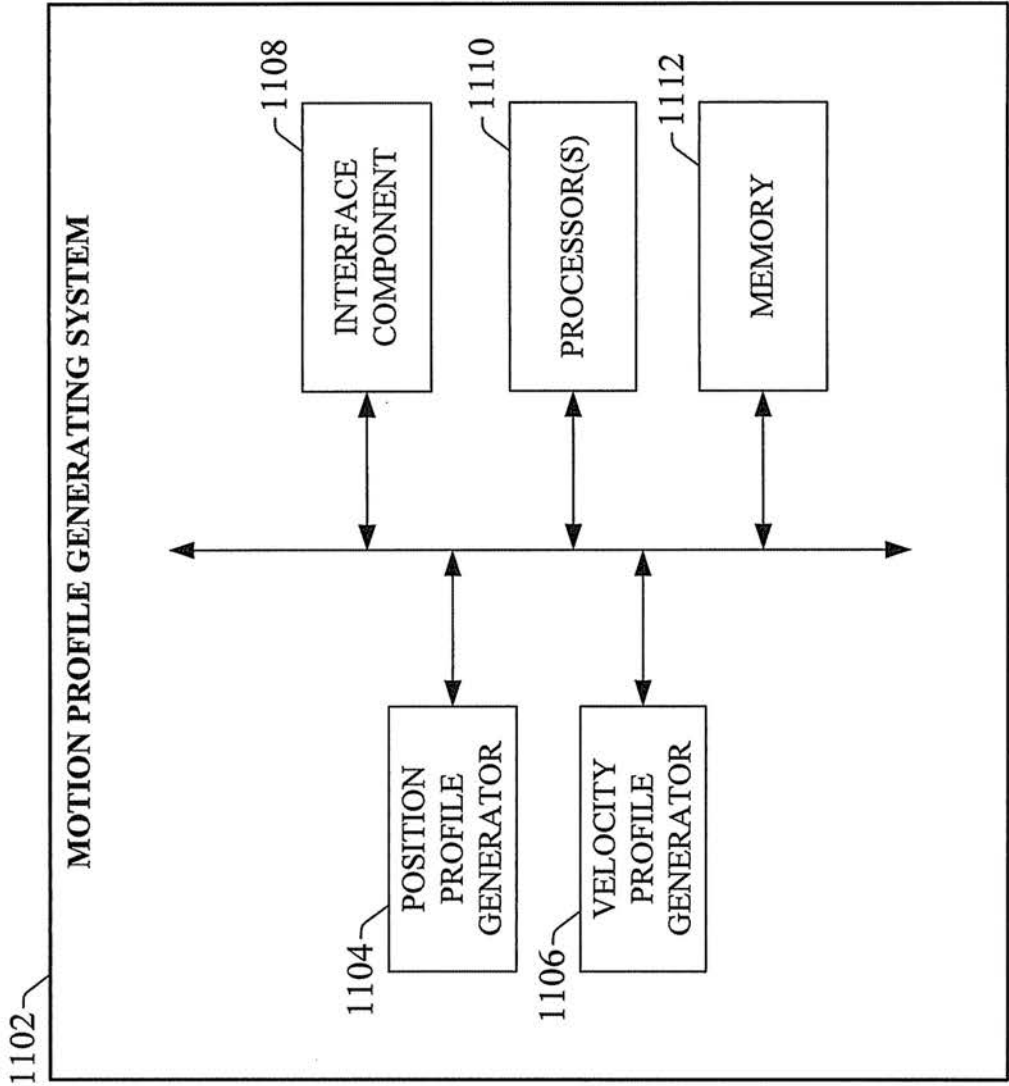
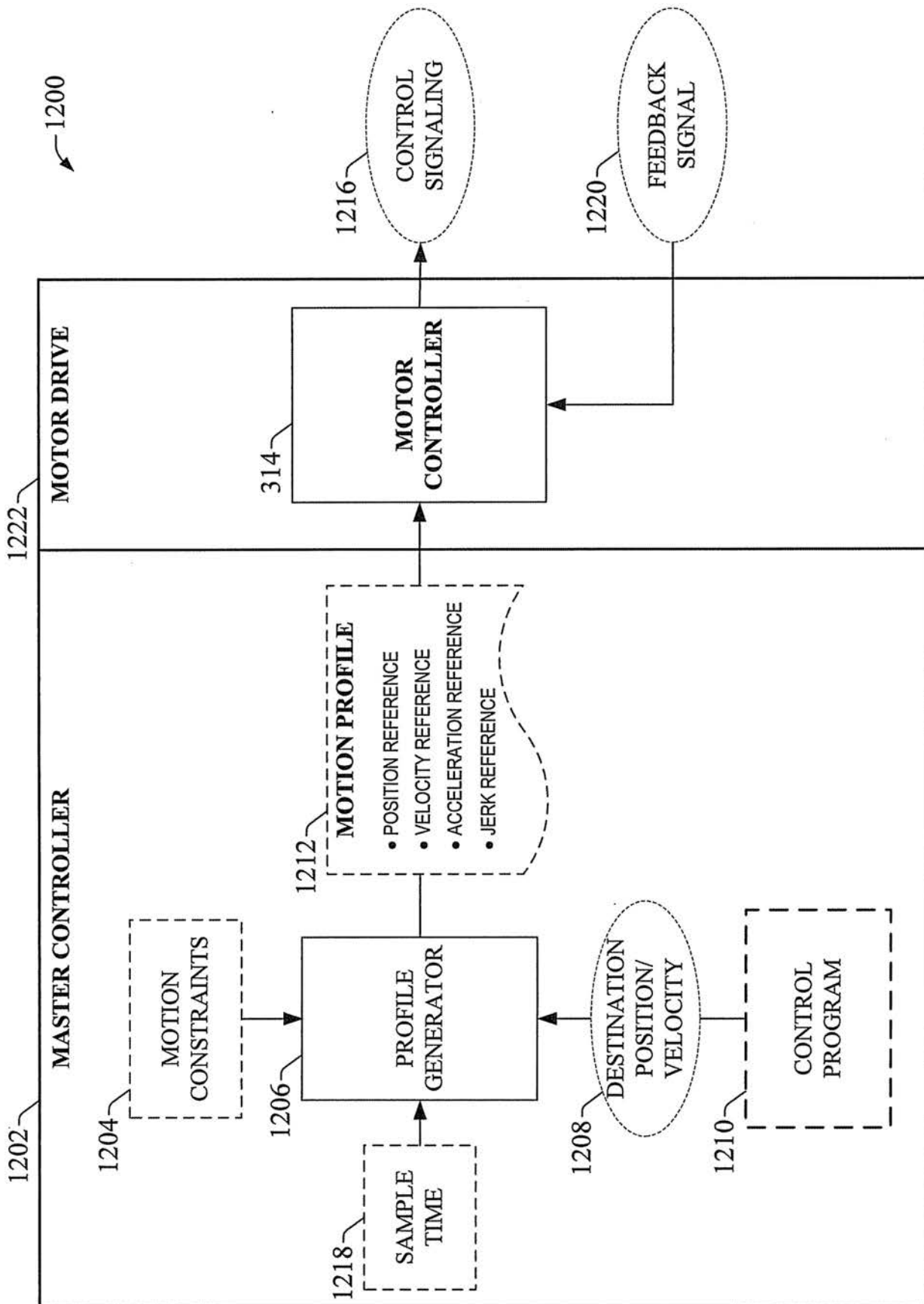
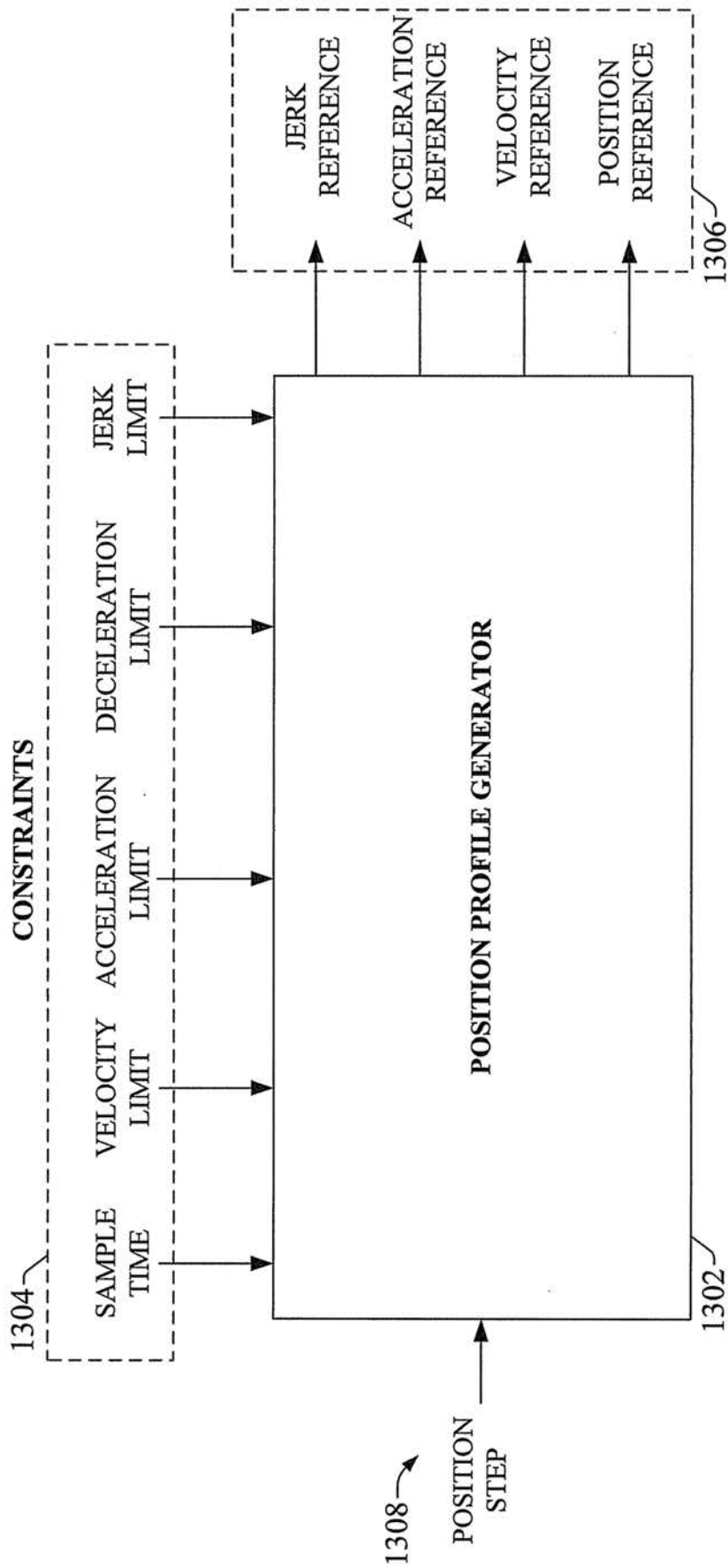


FIG. 11

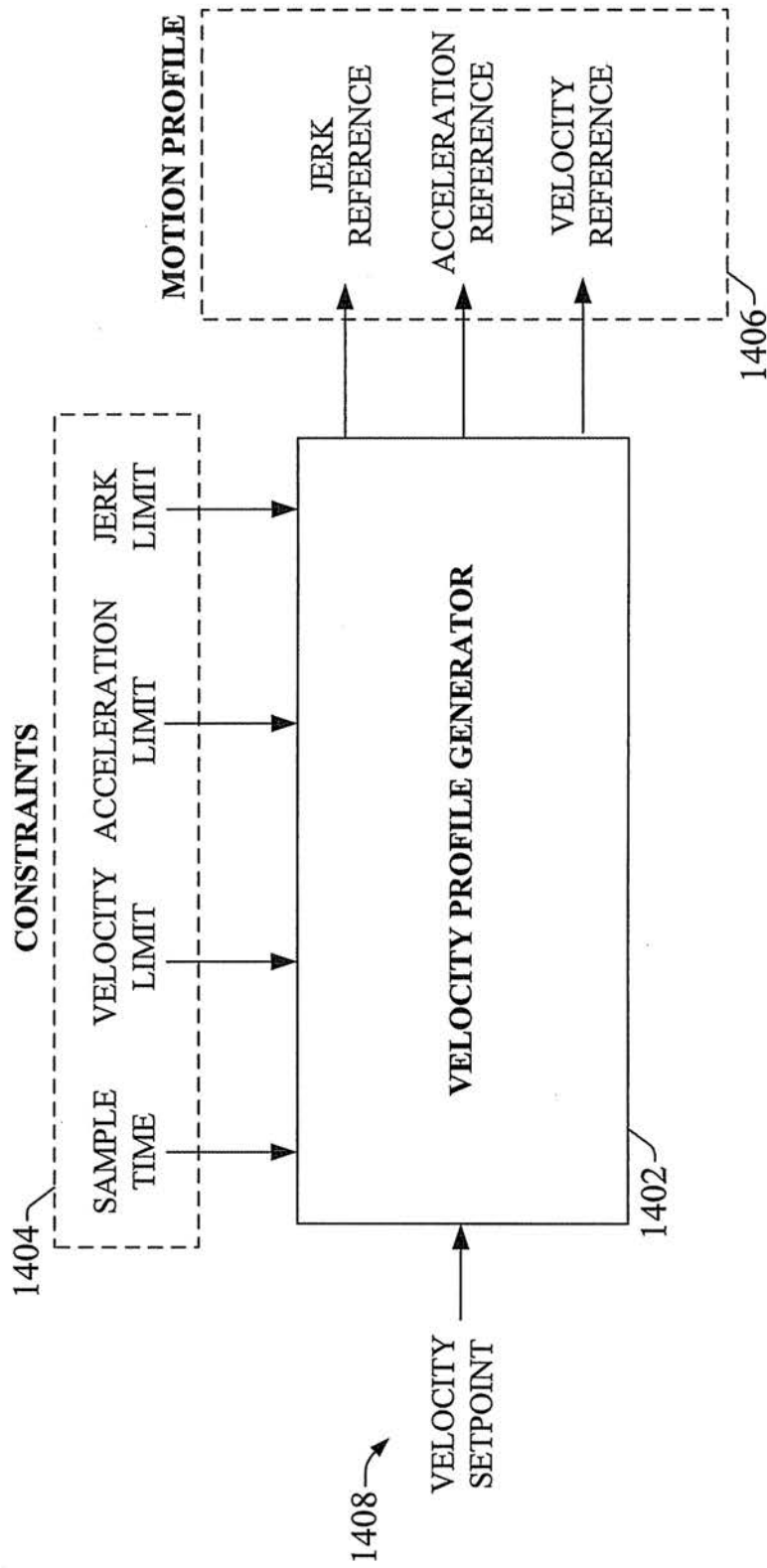
12/22

**FIG. 12**

13/22

**FIG. 13**

14/22

**FIG. 14**

15/22

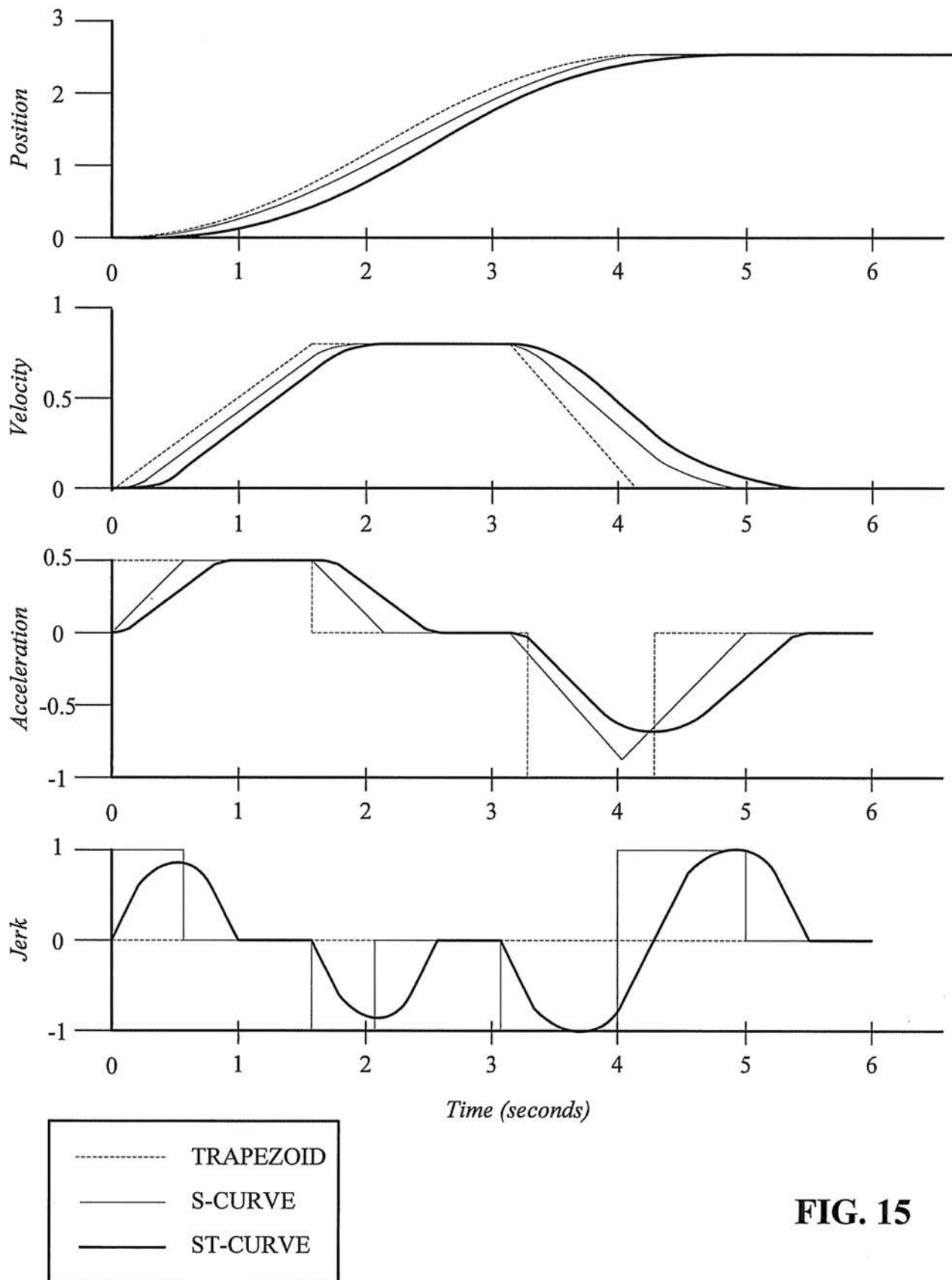


FIG. 15

16/22

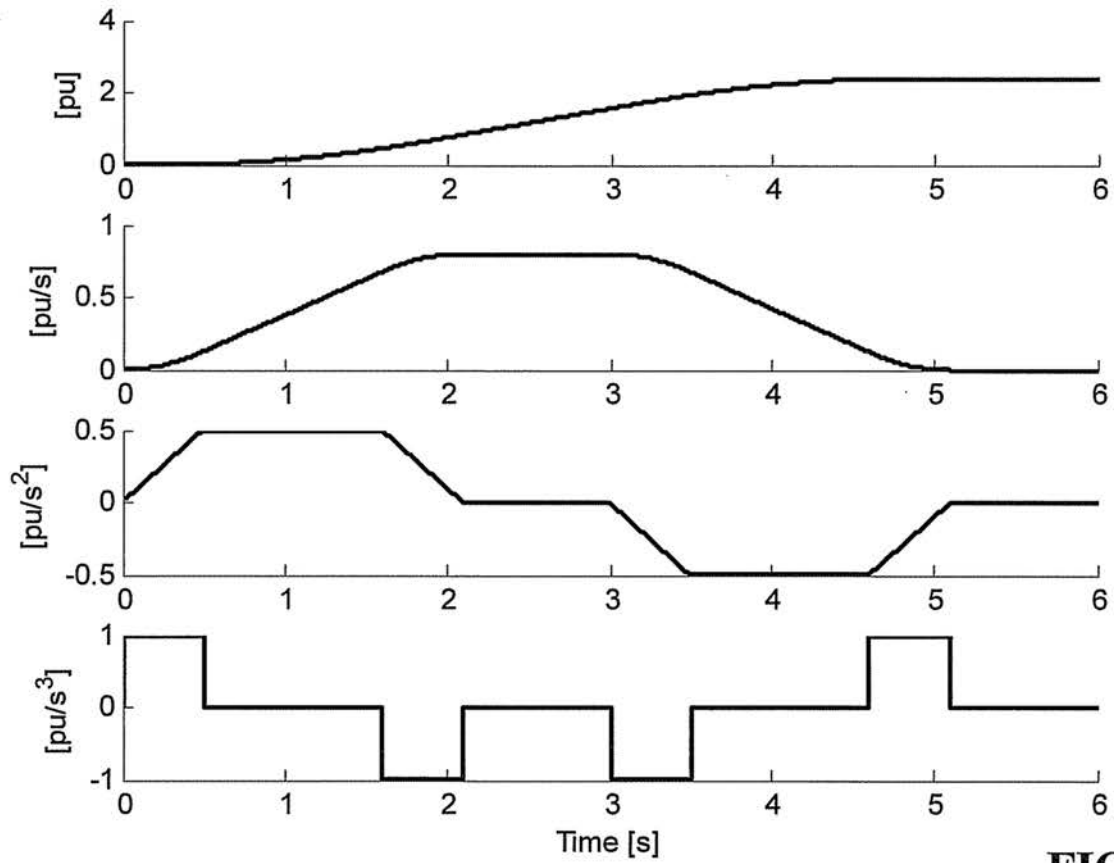


FIG. 16

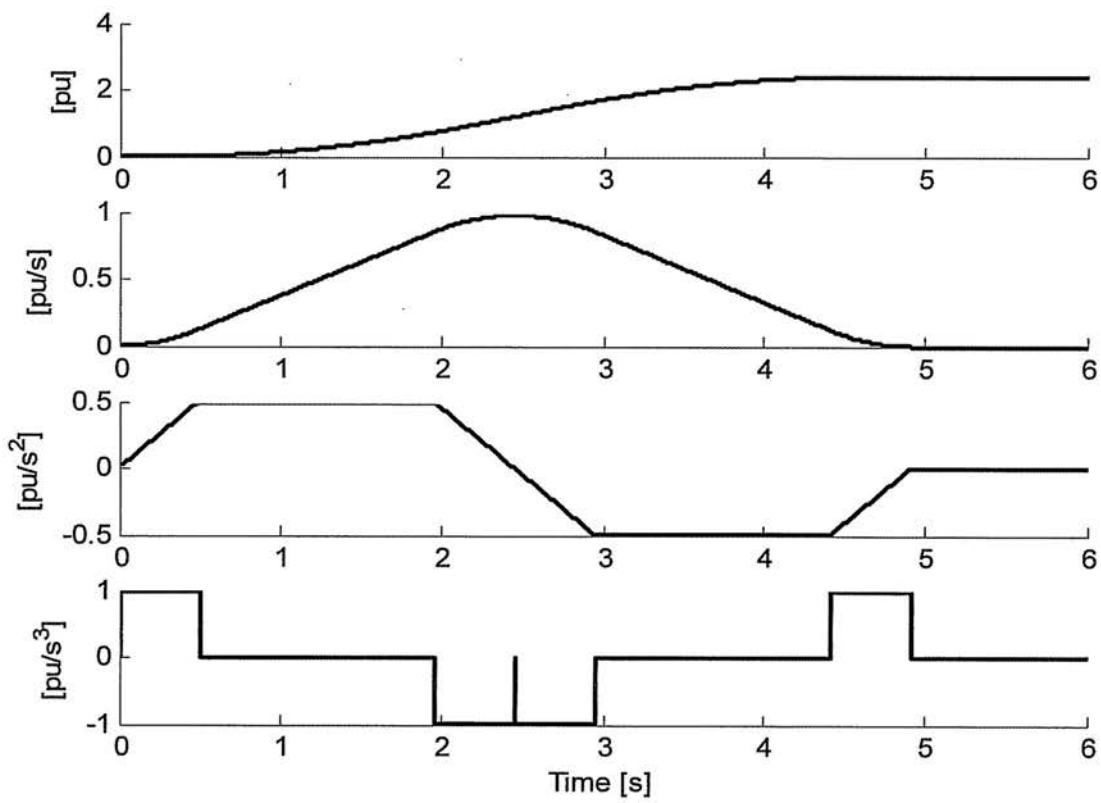


FIG. 17

17/22

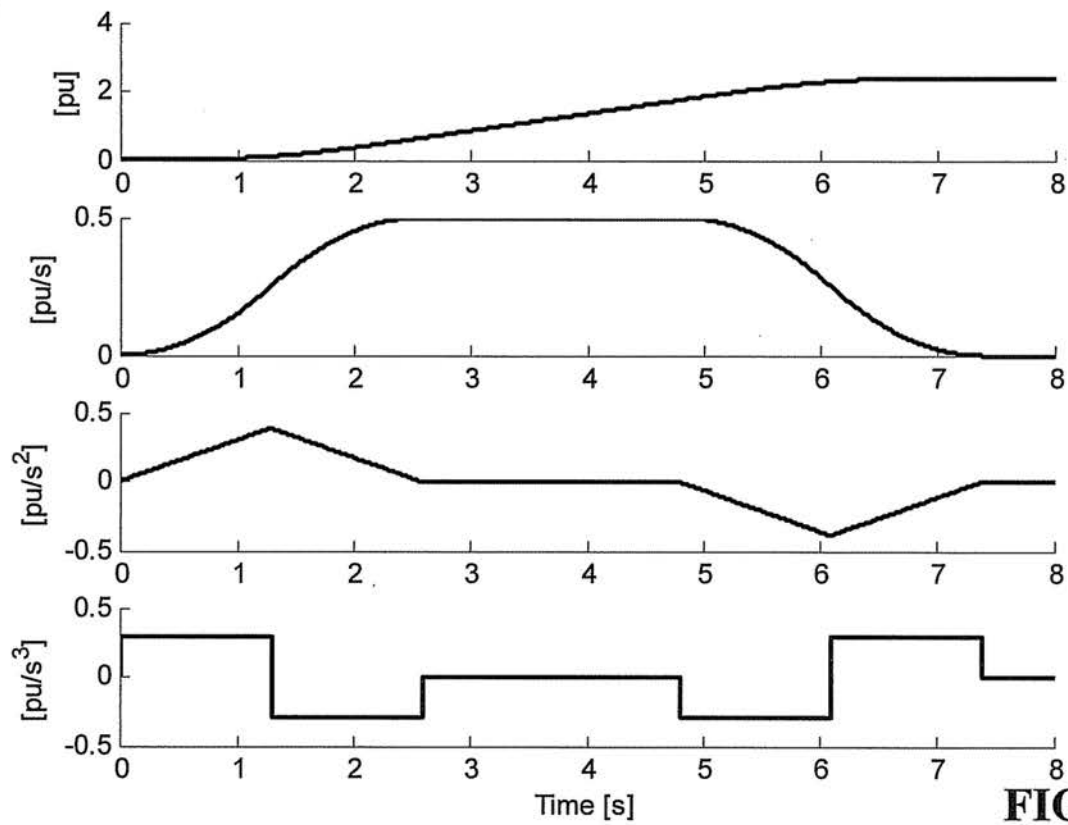


FIG. 18

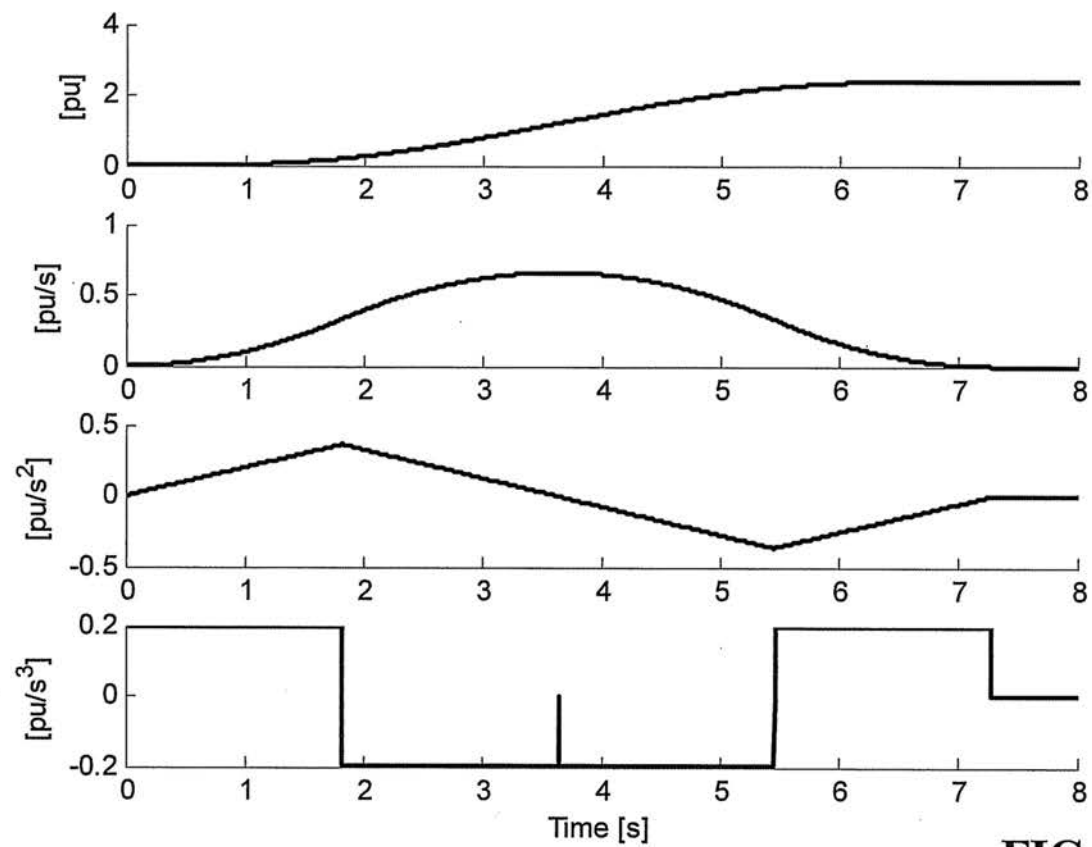
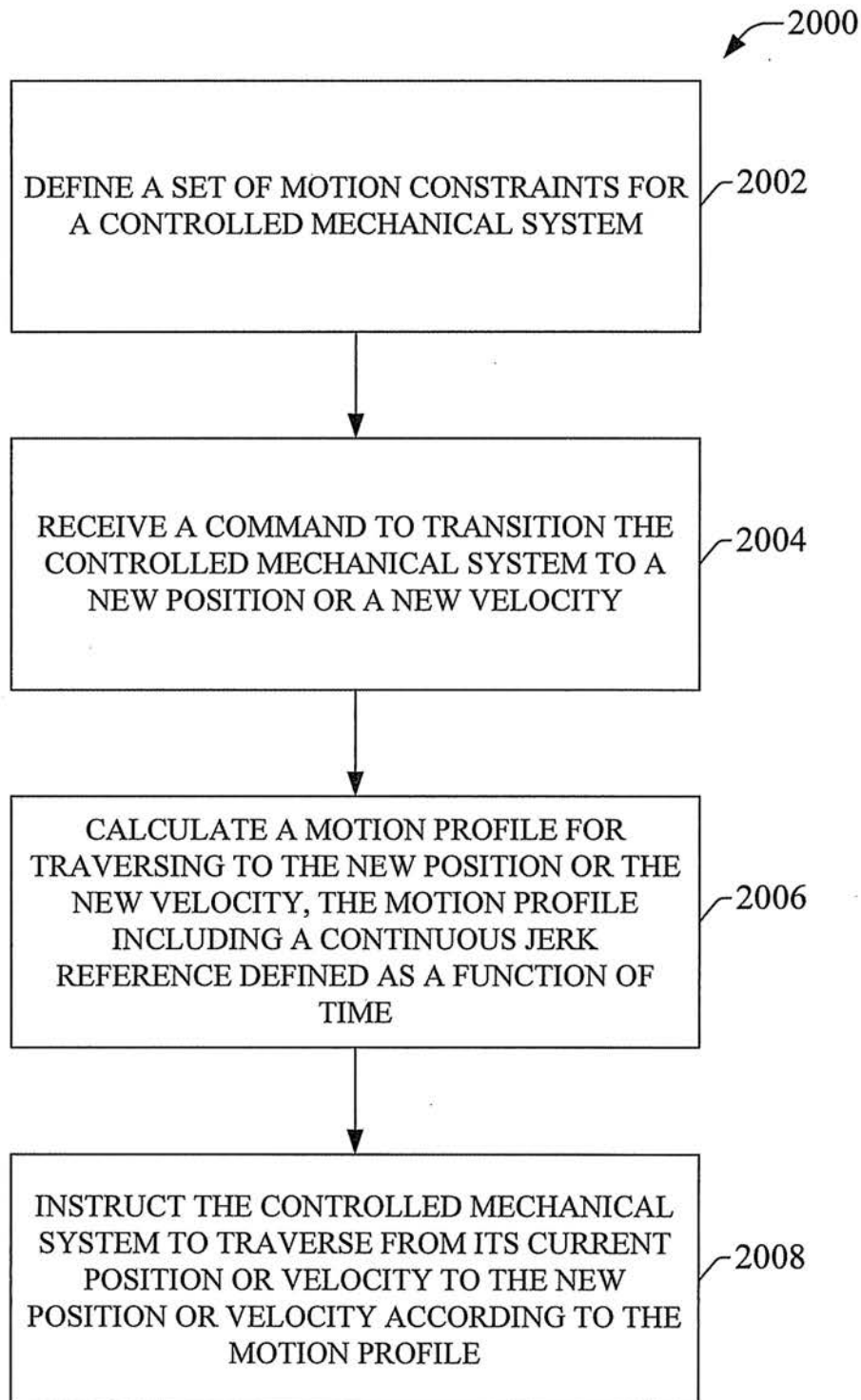


FIG. 19



18/22

**FIG. 20**

19/22

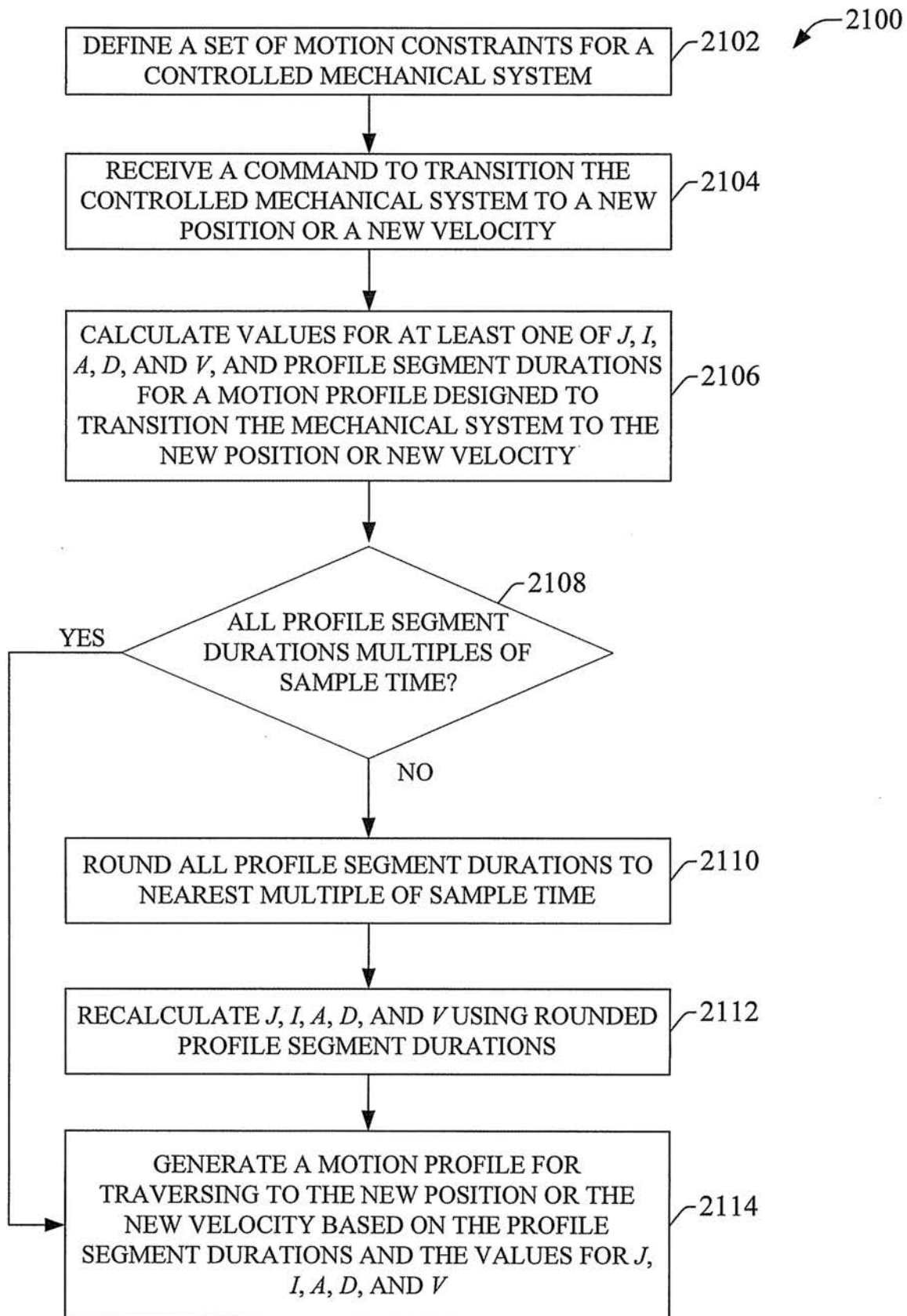
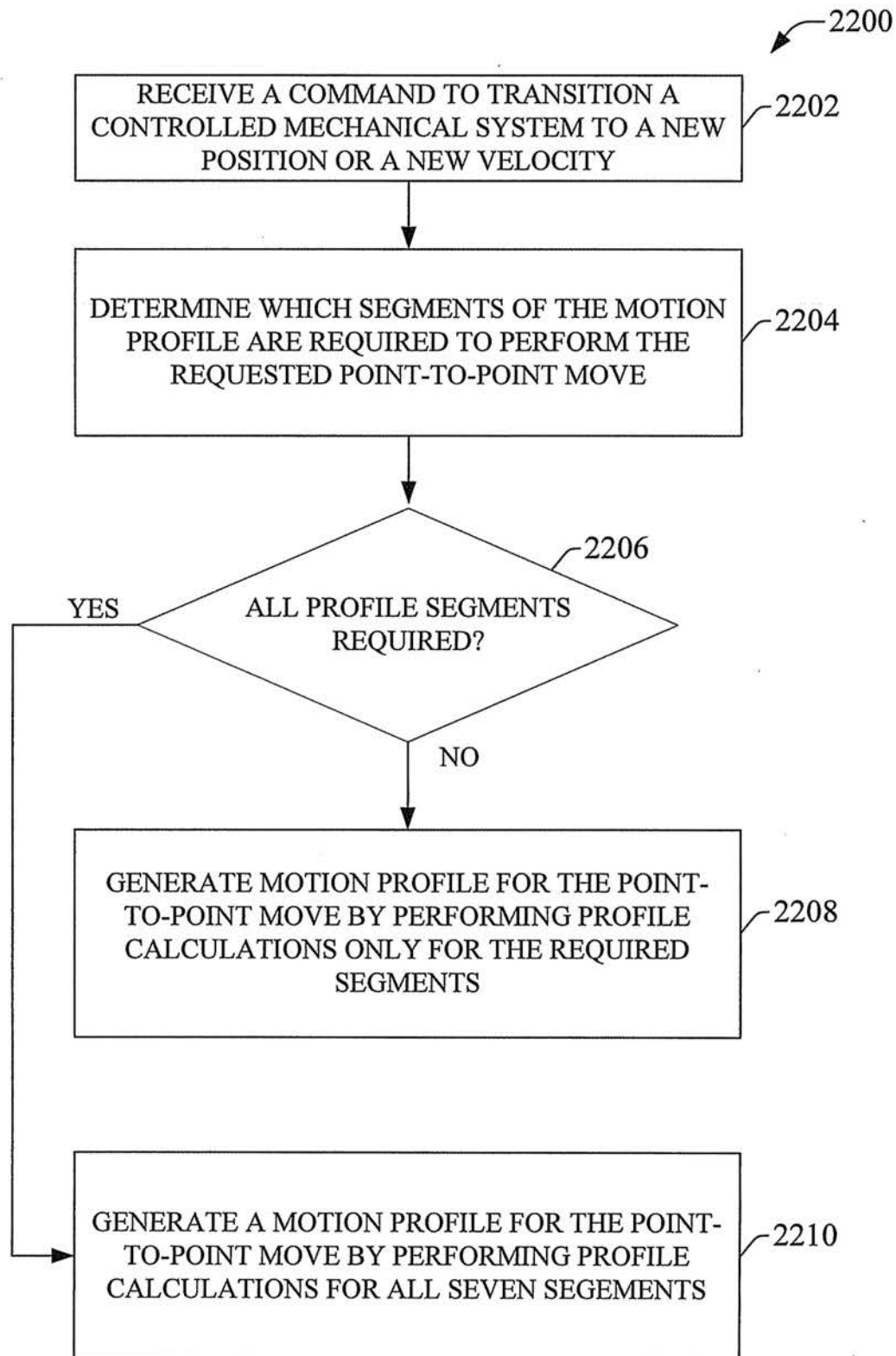
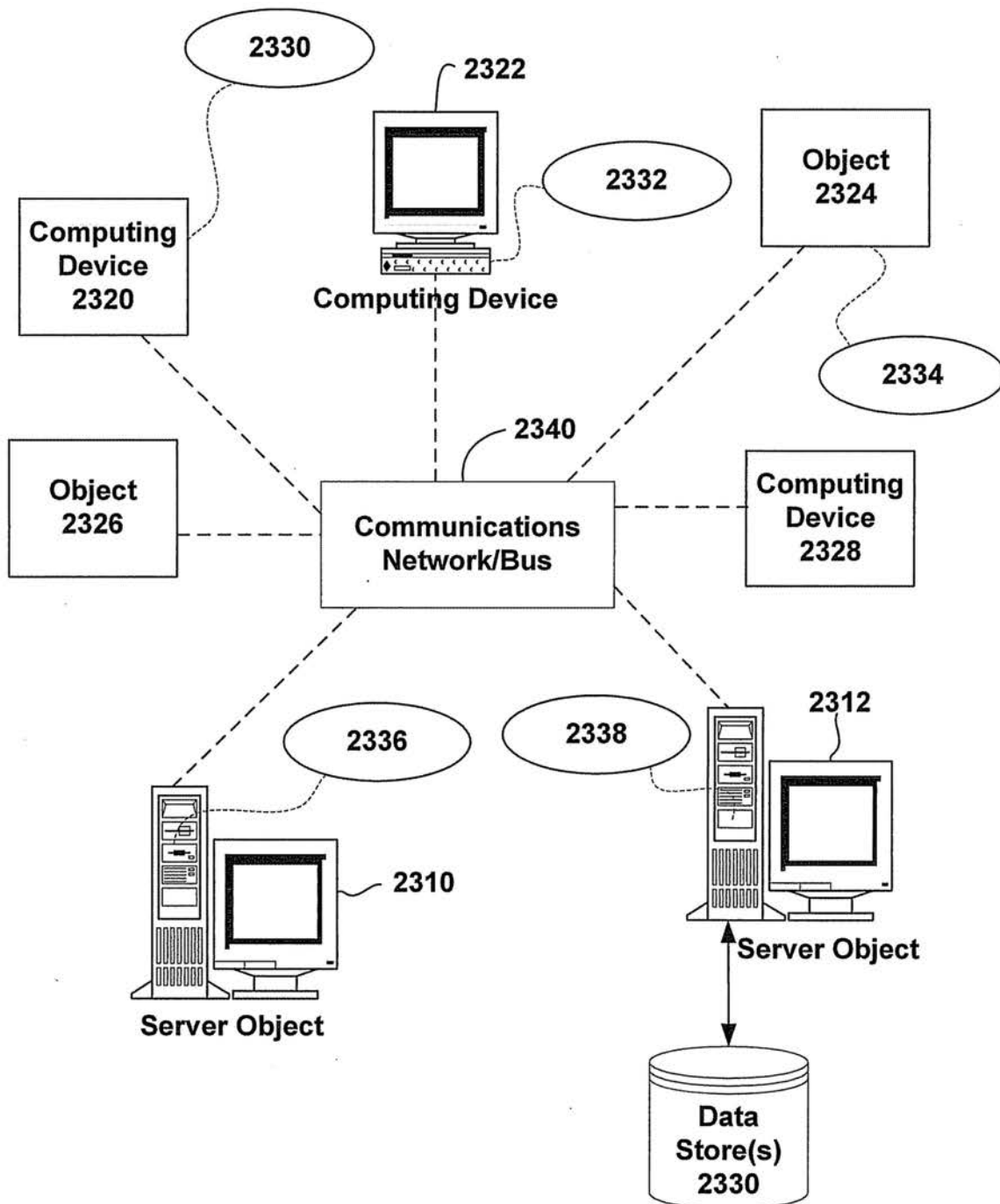


FIG. 21

20/22

**FIG. 22**

21/22

**FIG. 23**

Computing Environment 2200

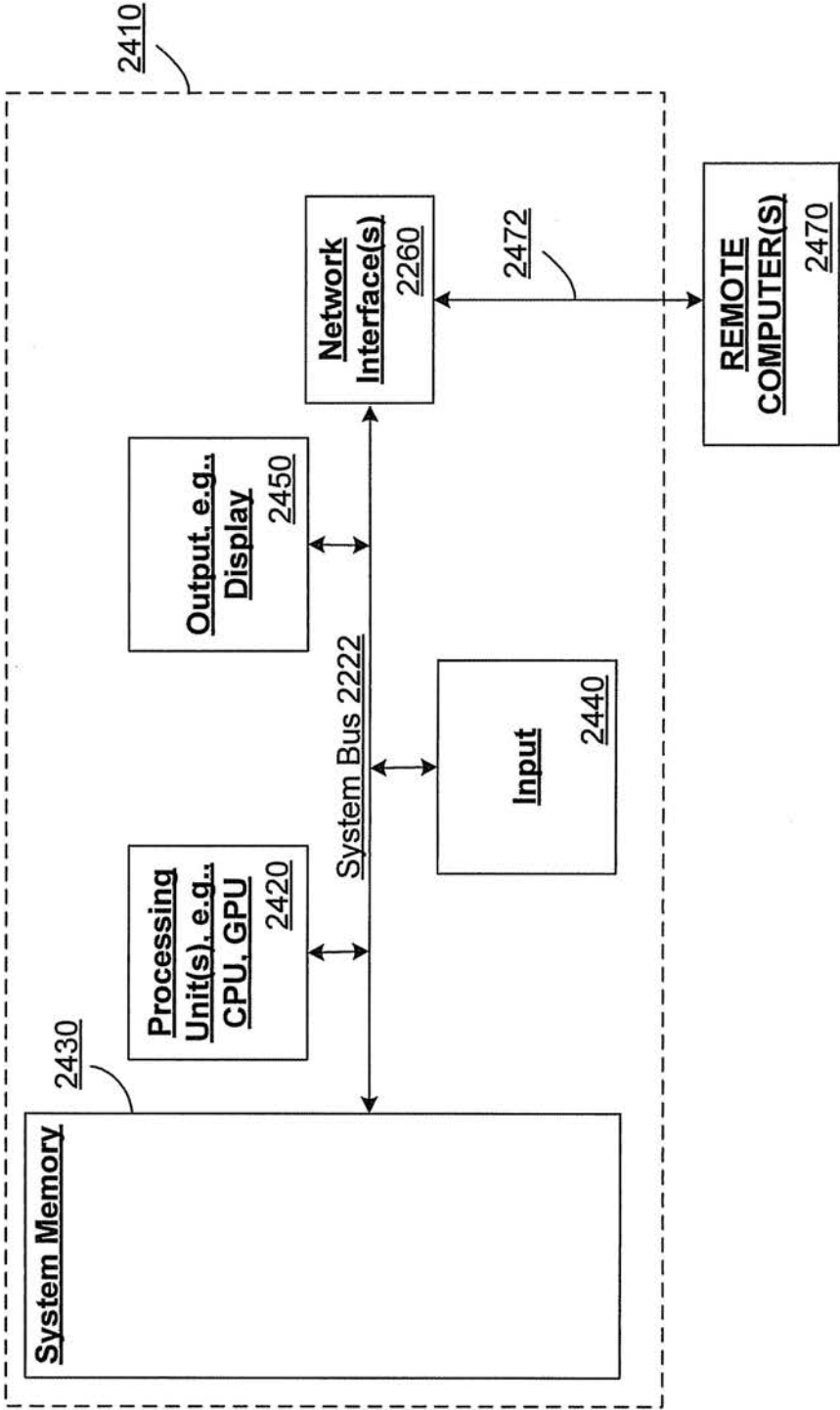


FIG. 24