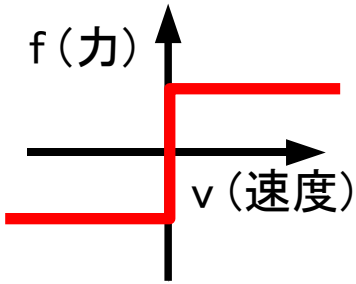
滑らかでないシステムの離散時間表現とヒューマンインタフェースへの応用

菊植 亮（Ryo Kikuuwe）

九州大学工学研究院
知能機械システム部門

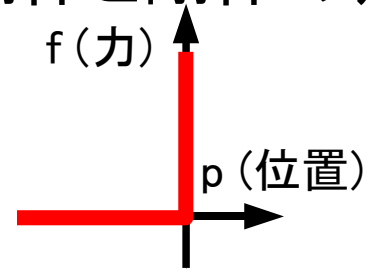
「滑らかでないシステム」とは

- 不連続な微分方程式に支配されるシステム．典型例としては・・・
  - 完全塑性（perfect plasticity）＝ クーロン摩擦特性



$$M\dot{v} \begin{cases} = Fv/|v| & \text{if } v \neq 0 \\ \in [-F, F] & \text{if } v = 0 \end{cases}$$

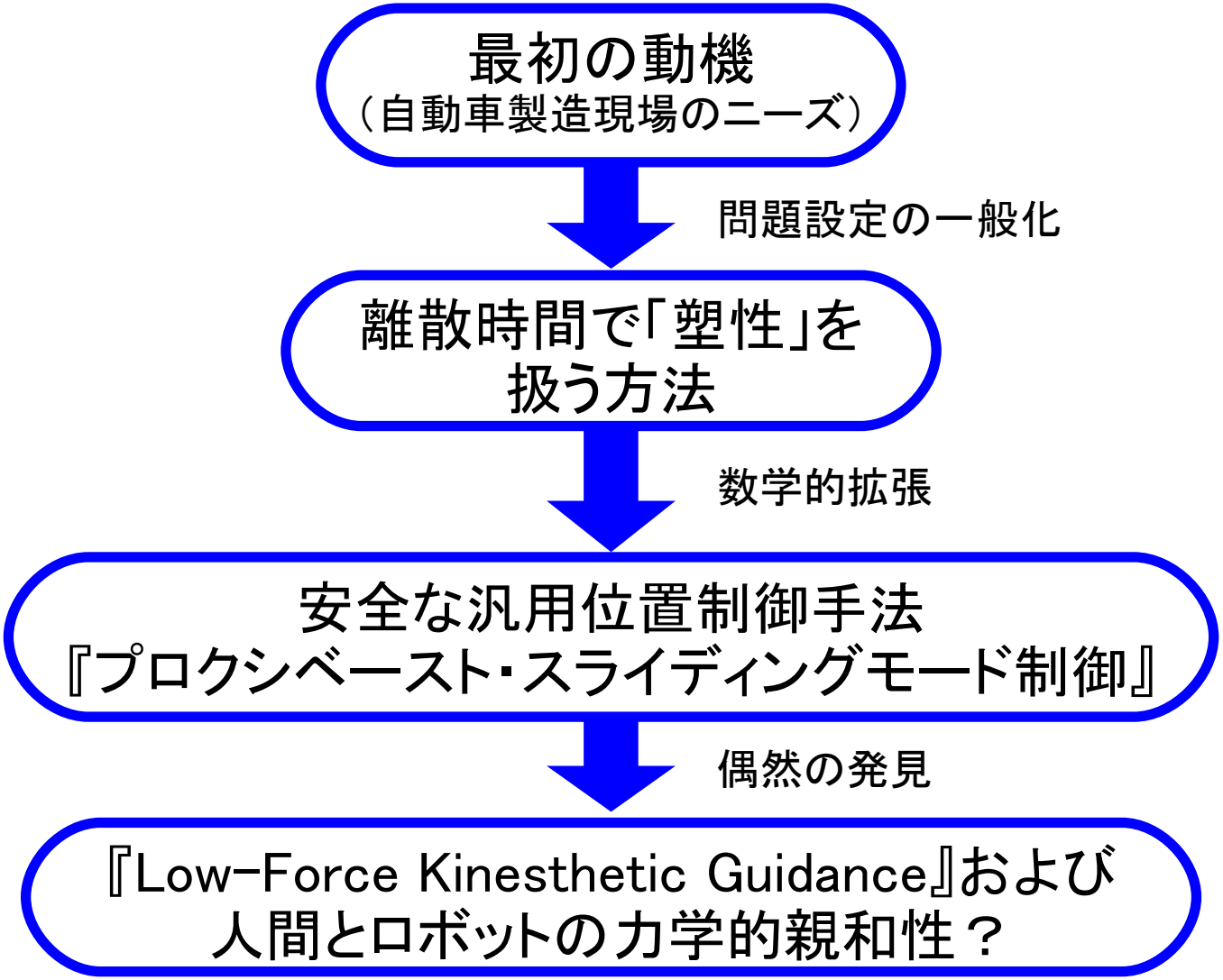
  - 剛体と剛体の片側接触



$$M\ddot{p} \begin{cases} = 0 & \text{if } p < 0 \\ \in [0, \infty) & \text{if } p = 0 \end{cases}$$

- 数学的に扱いづらいが，日常的にはありふれている．
- シミュレーションや仮想現実に適切に組み込む必要がある．
- うまく作れば／使えば，色々と役に立つ（？）

本日の内容


最初の動機
（自動車製造現場のニーズ）
問題設定の一般化
離散時間で「塑性」を
扱う方法
数学的拡張
安全な汎用位置制御手法
『プロクシベースト・スライディングモード制御』
偶然の発見
『Low-Force Kinesthetic Guidance』および
人間とロボットの力学的親和性？

最初の動機

自動車産業からのニーズ

- 製造・組み立て現場における重量物搬送と精密位置決め作業をアシストするロボットがほしい.
- 重量物を動かすこと自体に技術的問題はない.
- ポイントは,**「作業者にどのように反力を与えると位置決め効率がアップするか？」**

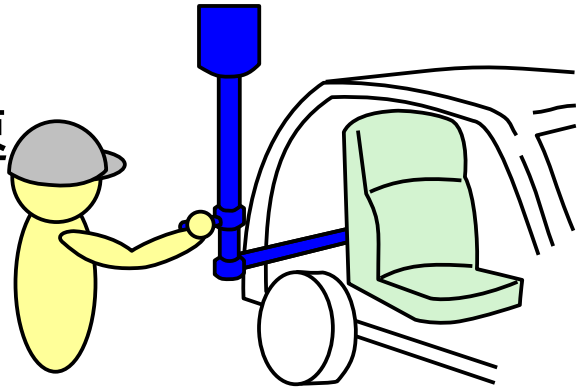
⇒ インピーダンス制御？

- 作業者からの力と搬送物の運動(加速度・速度)が, 何らかの式, たとえば

$$h = M\ddot{p} + B\dot{p}$$

を満たすようにロボットを制御する.

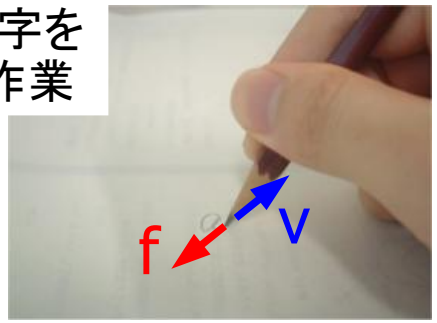
- しかし, 線形で固定のインピーダンスでは, 満足な反力が得られない.(移動中に重く感じるが, 位置決めではユラユラして頼りなくなる.)

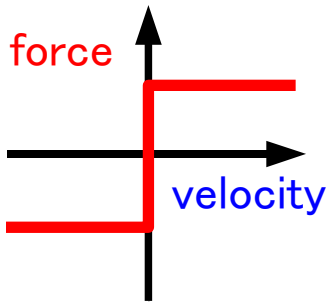


アイデア:「クーロン摩擦特性」すなわち「塑性」

- 塑性とは
  - 降伏力(静止摩擦力)以下の力では外力と平衡する.
  - 降伏力より大きい力で不可逆な変位(塑性変位)を生じる.
- 最も身近なダイナミクス
- 日常の精密作業に不可欠(手ブレ, 外乱を除去)

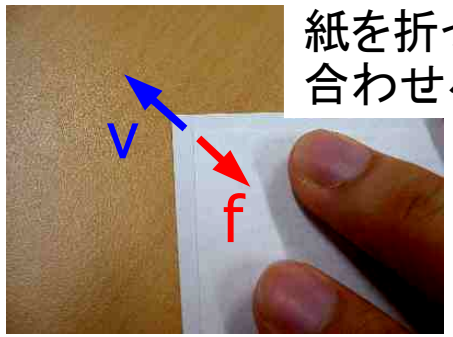
紙に字を書く作業





$$f = F\mathrm{sgn}(v)$$

紙を折って角を合わせる作業

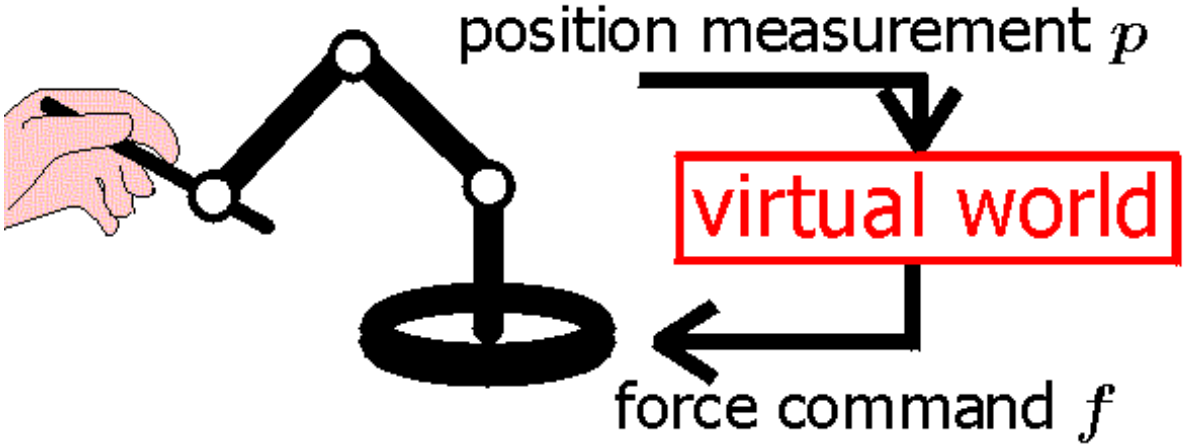


- **技術的問題**:不連続であるので, 離散時間のインピーダンス制御系による実現(シミュレート)が困難.

⇒ 離散時間における不連続性の取り扱い全般に(個人的に)興味が移行し, 現在に至る.

離散時間における「塑性」のアドミッタンス表現とインピーダンス表現

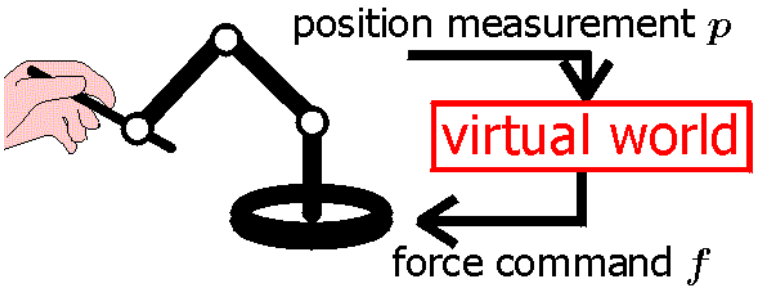
力覚(ハプティック)フィードバック



- デバイスと接触している操作者と，そのデバイスとの境界面において，なんらかの位置と力の関係を実現すること.
- 応用例:
  - 作業支援,
  - バーチャルリアリティ(医療訓練・設計など)
  など

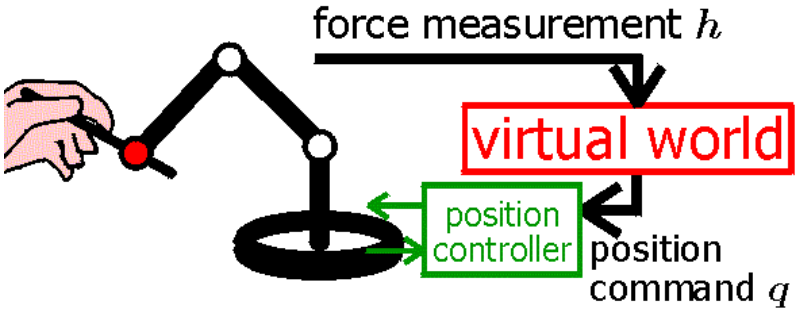
力覚フィードバックにおける入出力

- インピーダンス型：位置入力・力出力
  - デバイスにはバックドライバビリティが必要



- Sensable Technologies 社:『PHANTOM』
  - http://www.sensable.com/
- CyVerse社:『SPIDAR』
  - http://www.cyverse.co.jp/Products/spidar.html

- アドミッタンス型：力入力・位置出力
  - デバイスには力センサが必要



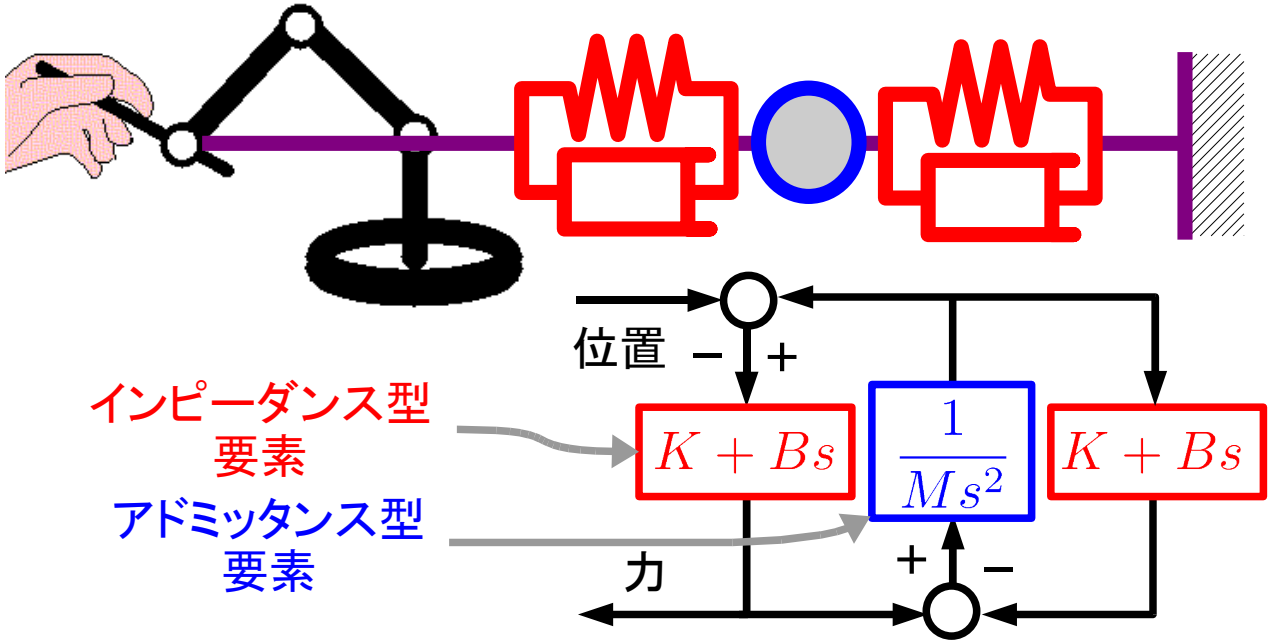
- Moog FCS社:『HAPTIC MASTER』
  - http://www.fcs-cs.com/robotics/products/hapticmaster
- John Hopkins Univ.:『Steady-Hand Robotic System』
- Munich Tech. Univ.:『ViSHaRD10』

- どちらを使うべきかはデバイスの特性で決まる.
- いずれにしても，位置・力の入出力の確定が必要.

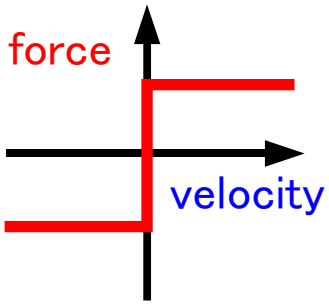
仮想世界の構成要素の入出力

- 仮想世界の構成要素は，
  - インピーダンス型の入出力表現（位置⇒力），or
  - アドミッタンス型の入出力表現（力⇒位置）

  のどちらかを持つと便利である．
- 仮想世界全体は，これらの構成要素のネットワークとして表現される．



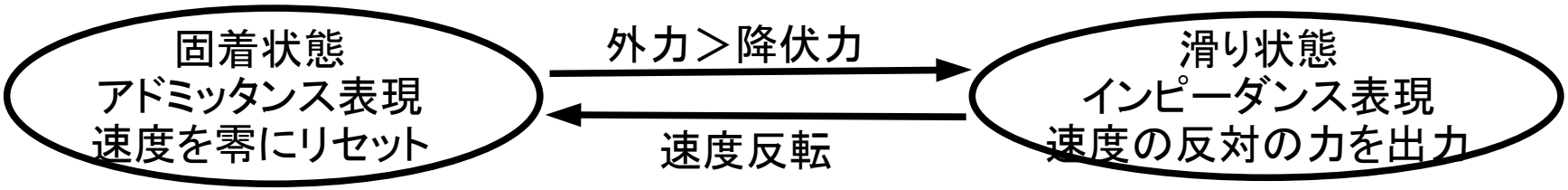
「塑性」の入出力表現は？



- 「インピーダンス」（速度⇒力）とすると
  - 速度が非零 ⇒ 力は速度と反対
  - 速度が零 ⇒ 力が決まらない.
- 「アドミッタンス」（力⇒速度）とすると
  - 力が降伏力未満 ⇒ 速度は零
  - 力が降伏力以上 ⇒ 速度が決まらない.

⇒ 力と速度の入出力関係を決められない.

- 状態によって入出力を切り替えることは可能だが・・



× 入出力の反転は他の構成要素との関連にも影響する
⇒ プログラムが複雑になり，再利用性も低下.

× 多次元に拡張しにくい（「速度反転」が定義できない）

従来の考え方

- 速度に閾値を設けて，零速度を区別するモデル（Karnoppモデルなど）が主流
  - 閾値以上の速度では速度と反対向きの力を出力
  - 閾値以下の速度では
    - 速度に比例する力を出力（粘性で近似）
    - 外力とつりあう力を出力

    ・・・など

$$f = \begin{cases} F & \text{if } v > \varepsilon \\ ? & \text{if } v \in [-\varepsilon, \varepsilon] \\ -F & \text{if } v < -\varepsilon \end{cases}$$

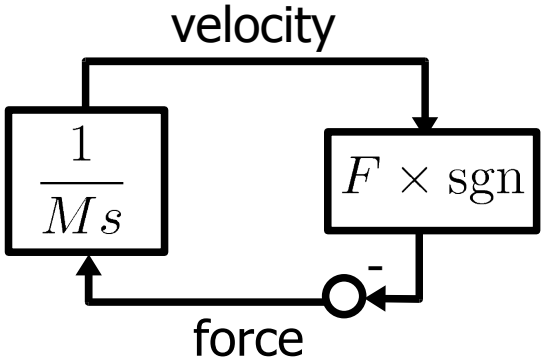
× 閾値の選び方に敏感になる.
  - 閾値が小さすぎると⇒ 零速度交差の繰り返しで力が振動（**チャタリング**）.
  - 閾値が大きすぎると⇒閾値以下の速度で**ドリフト.**

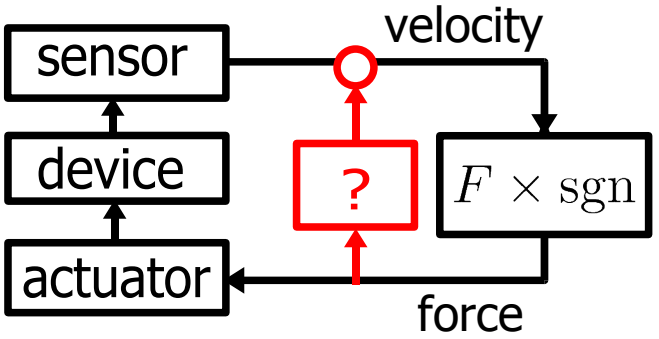
⇒ **チャタリング**か**ドリフト**のどちらかが発生.

× そもそも閾値に物理的な意味がない.

離散時間で問題が生じる原因

- 離散時間ではチャタリングが起こる.
- しかし，連続時間ではチャタリングが起こらない.

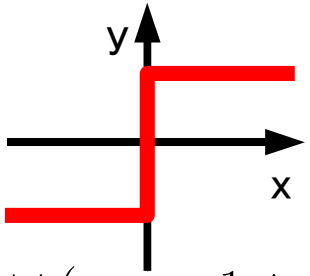




- 原因：符号関数を囲む最も内側のフィードバックループに時間遅れがあると，チャタリングが生じる.
- 解決策： 制御器の中に時間遅れのないフィードバックループを作ればよい.

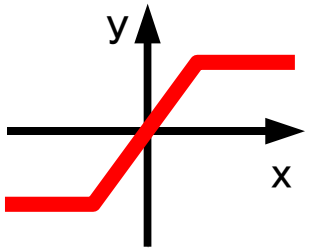
時間遅れの無いフィードバック

- 符号関数: $\mathrm{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ [-1, 1] & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$

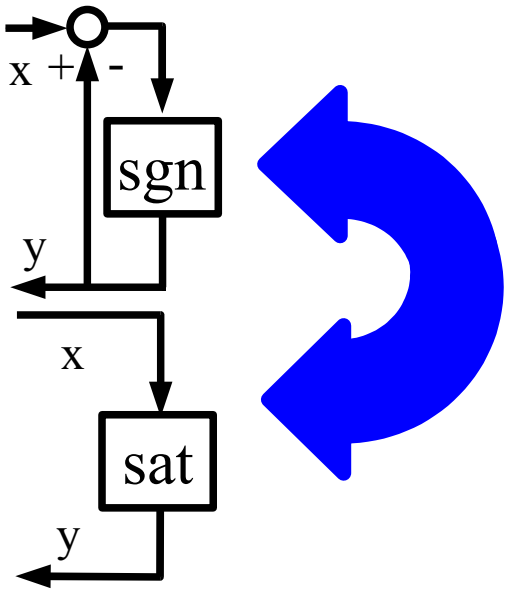


$$y = \mathrm{sgn}(x) \iff (y = 1 \wedge x > 0) \vee (-1 < y < 1 \wedge x = 0) \vee (y = -1 \wedge x < 0)$$

- 単位飽和関数: $\mathrm{sat}(x) = \begin{cases} 1 & \text{if } x > 1 \\ x & \text{if } x \in [-1, 1] \\ -1 & \text{if } x < -1 \end{cases}$



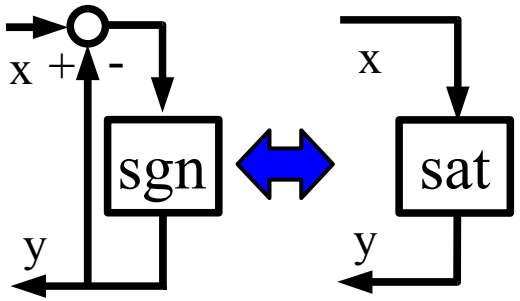
$$y = \mathrm{sgn}(x - y)$$
$$\iff (y = 1 \wedge x - y > 0)$$
$$\vee (-1 < 0 < 1 \wedge x - y = 0)$$
$$\vee (y = -1 \wedge x - y < 0)$$
$$\iff y = \mathrm{sat}(x)$$



等価

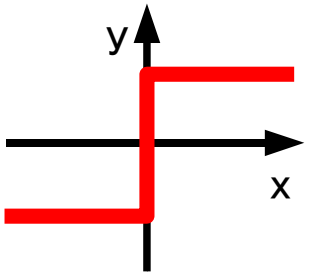
定理

- 下記の2式は等価である.
  - $y = \mathrm{sgn}(x - y)$
  - $y = \mathrm{sat}(x)$

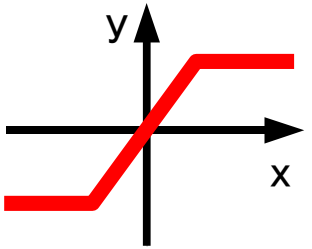


ただし，ここで，

sgnは符号関数：
$$\mathrm{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ [-1, 1] & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$



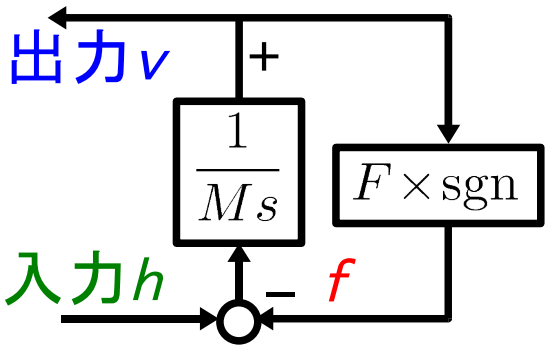
satは単位飽和関数：
$$\mathrm{sat}(x) = \begin{cases} 1 & \text{if } x > 1 \\ x & \text{if } x \in [-1, 1] \\ -1 & \text{if } x < -1 \end{cases}$$

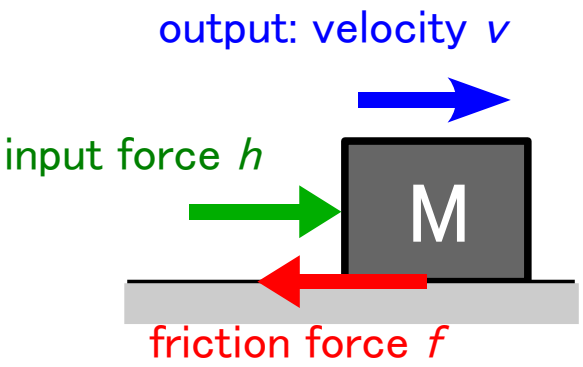


- **符号関数を時間遅れのないフィードバックで閉じると，飽和関数に変換できる.**
- 時間遅れの無いフィードバック＝入力と出力の代数拘束.

塑性のアドミッタンス表現

[Kikuuwe et al.: IEEE Trans. on Robotics, 2006]





- 連続時間表現（微分代数方程式）

$$M\dot{v} = h - f$$

$$f = F\mathrm{sgn}(v)$$

- 離散時間表現（代数方程式）

$$(v_k - v_{k-1})/T = h_k - f_k$$

$$f_k = F\mathrm{sgn}(v_k)$$

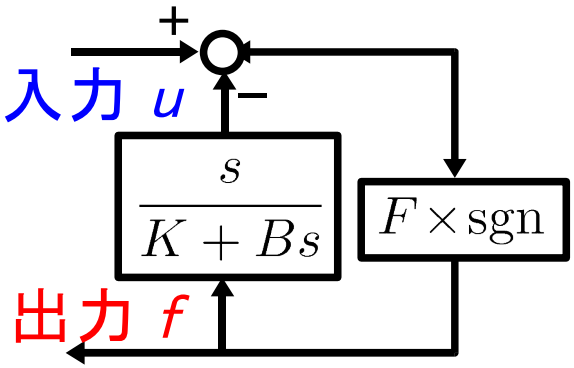
- 離散時間表現（上式を解く計算手順）

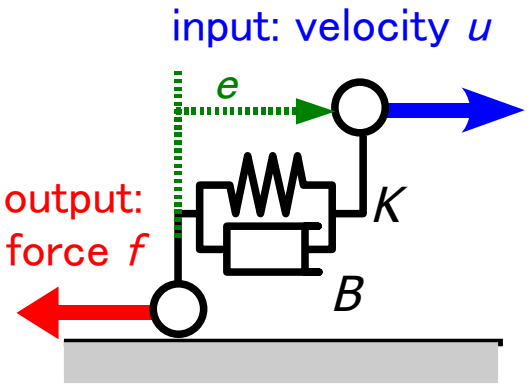
$$f_k := F\mathrm{sat}((Mv_k/T + h_k)/F)$$

$$v_k := v_{k-1} + T(h_k - f_k)/M$$

塑性のインピーダンス表現

[Kikuuwe et al.: IEEE Trans. on Robotics, 2006]





- 連続時間表現（微分代数方程式）

$$f = Ke + B\dot{e}$$

$$f = F\mathrm{sgn}(u - \dot{e})$$

- 離散時間表現（代数方程式）

$$f_k = Ke_k + B(e_k - e_{k-1})/T$$

$$f_k = F\mathrm{sgn}(u_k - (e_k - e_{k-1})/T)$$

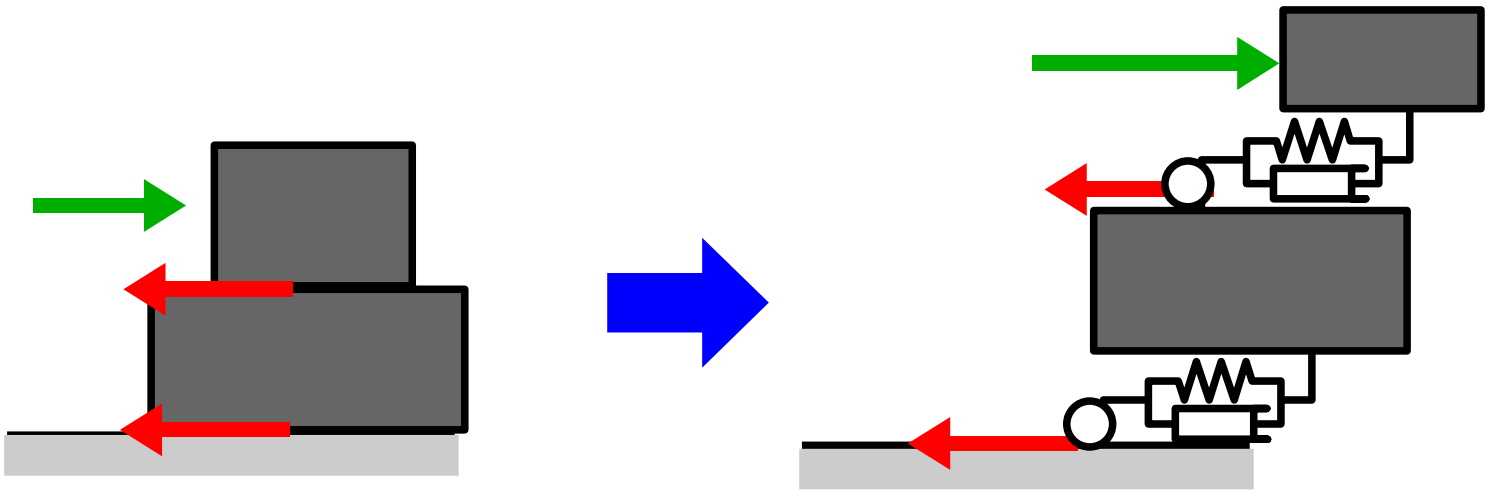
- 離散時間表現（上式を解く計算手順）

$$f_k := F\mathrm{sat}(((KT + B)u_k + Ke_{k-1})/F)$$

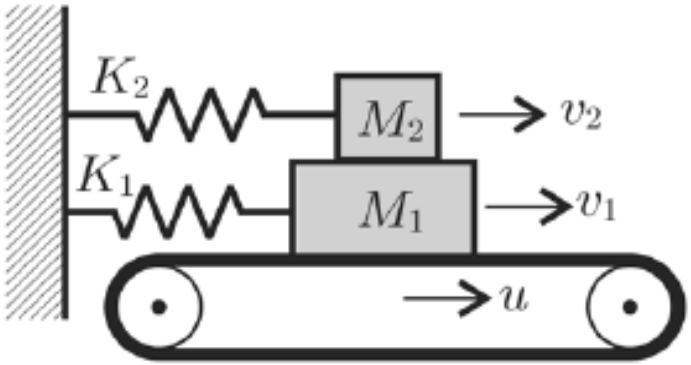
$$e_k := (Be_{k-1} + Tf_k)/(KT + B)$$

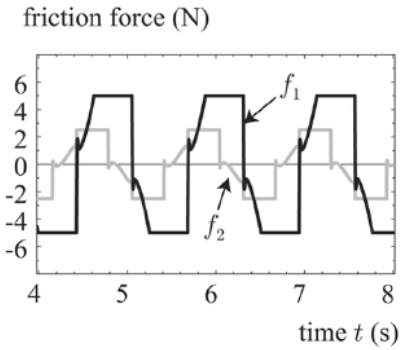
塑性のインピーダンス表現

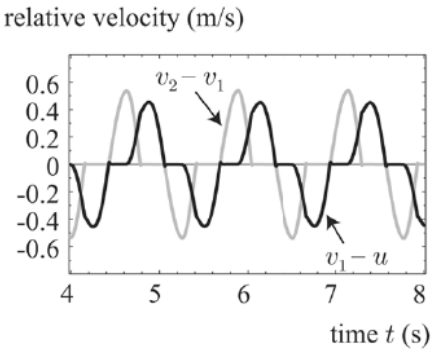
- すべての摩擦接触の表現に利用可能.
- 速度に閾値を設けるのではなく，仮想的な粘弾性要素をはさむ.
- 粘弾性要素の剛性と粘性は物性値を参考に or できるだけ大きい値に. ⇒ ドリフトは起こらない. 適切に選べばチャタリングもおこらない.
- 「ペナルティベース」手法の一種

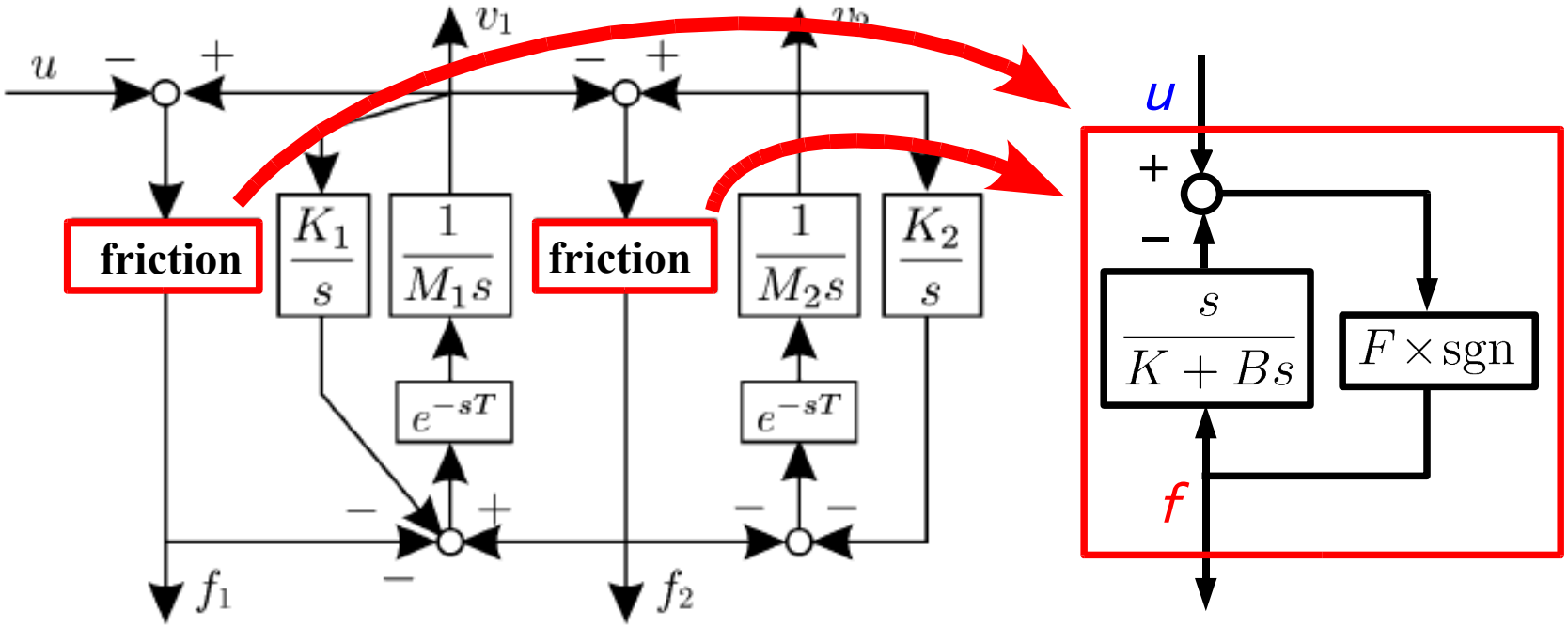


使用例


K_2
M_2
v_2
K_1
M_1
v_1
u

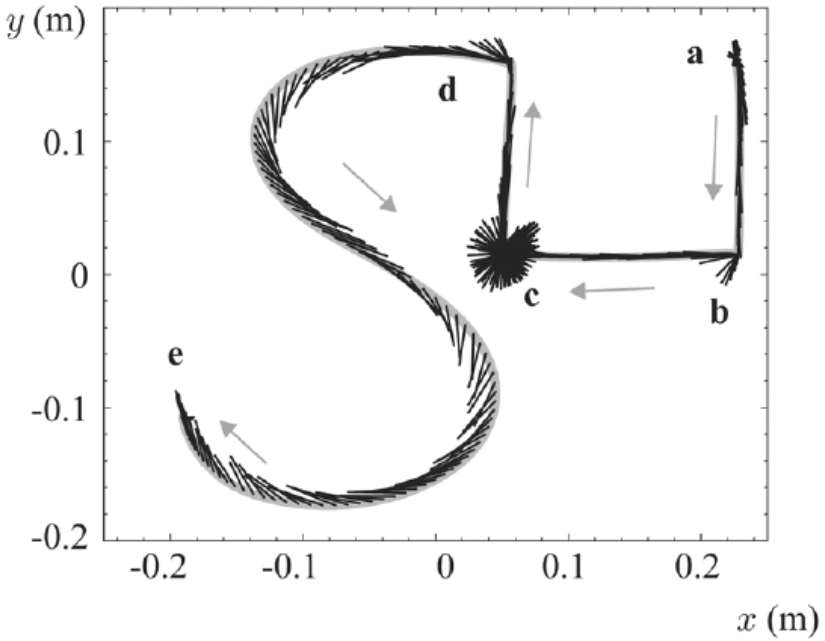

friction force (N)
6
4
2
0
-2
-4
-6
f_1
f_2
4
5
6
7
8
time t (s)

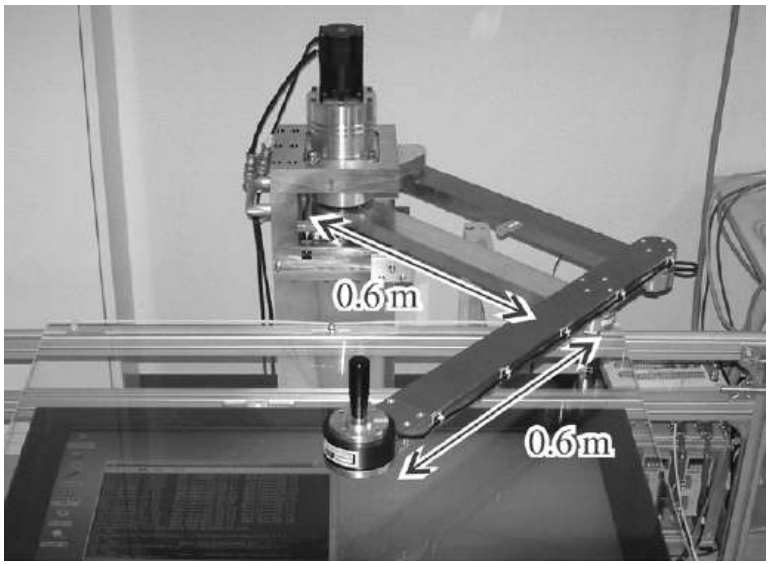

relative velocity (m/s)
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
$v_2 - v_1$
$v_1 - u$
4
5
6
7
8
time t (s)


u
v_1
v_2
friction
friction
$\frac{K_1}{s}$
$\frac{1}{M_1 s}$
$\frac{1}{M_2 s}$
$\frac{K_2}{s}$
e^{-sT}
e^{-sT}
f_1
f_2
u
$\frac{s}{K + Bs}$
$F \times \mathrm{sgn}$
f

等方性の摩擦

- 2次元・3次元の等方性の摩擦は，機械的に作るのが難しいが，アドミッタンス制御を用いれば可能.



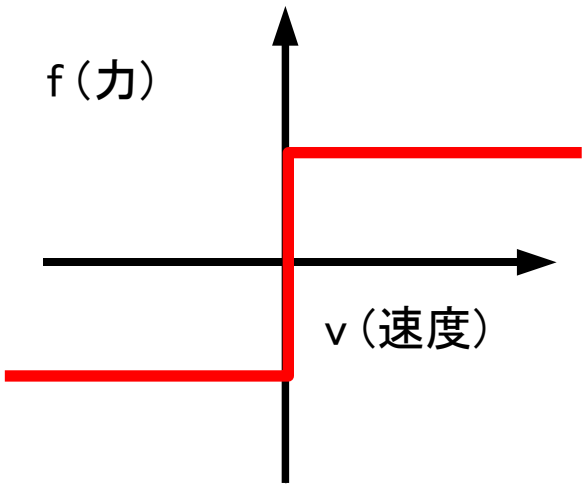


⇒ 精密位置決め作業の支援に有用？

安全なロボット制御

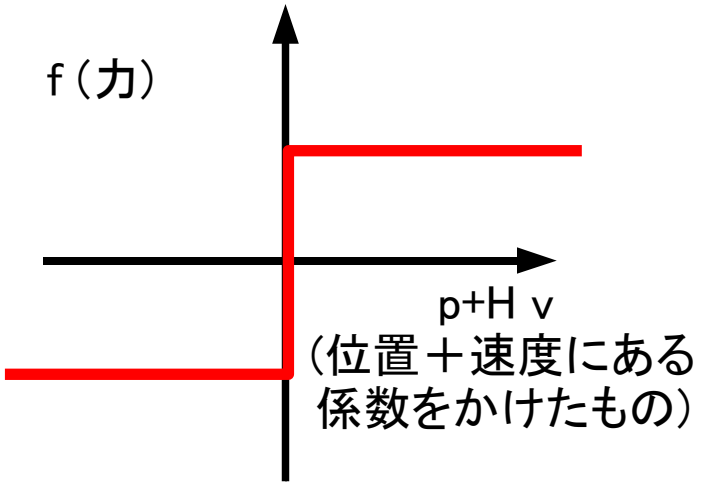
数学的な興味

- 完全塑性



$$f = -F\mathrm{sgn}(v)$$

- こう変更したらどうなる？



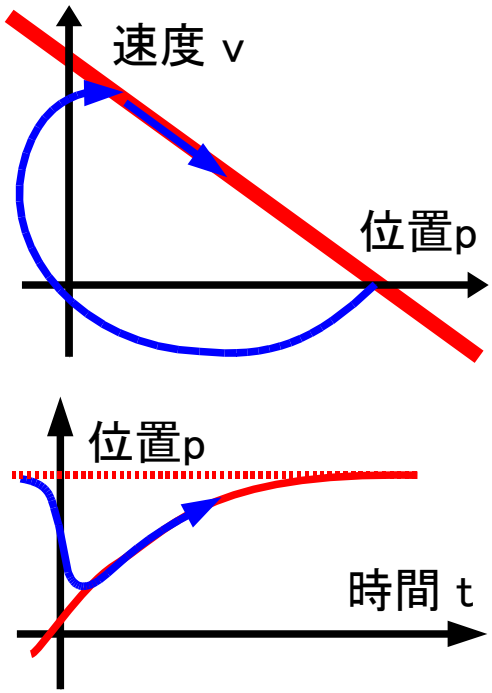
$$f = -F\mathrm{sgn}(-p - Hv)$$

- 自然界には（おそらく）存在しない，デジタル制御でしか作れない力学特性．

よく見れば，スライディングモード制御？

スライディングモード制御

- 状態空間内に設定したある「拘束多様体」に，制御対象の状態を拘束する.
- 「拘束多様体」を境界にして，操作量（アクチュエータの力）を不連続に切り替える.

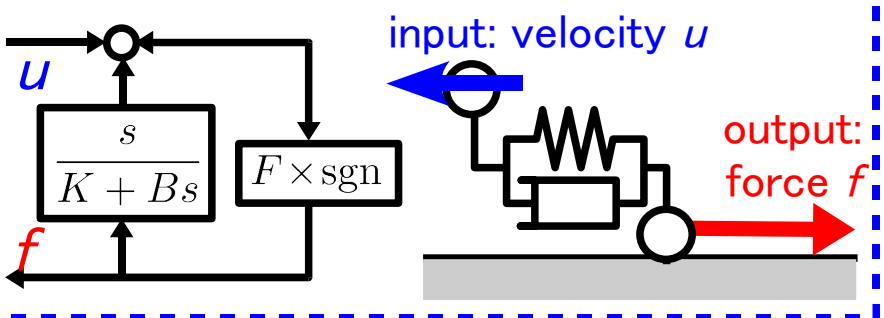


- もっとも単純な例：

$$f = F\mathrm{sgn}(p_d - p - H\dot{p})$$

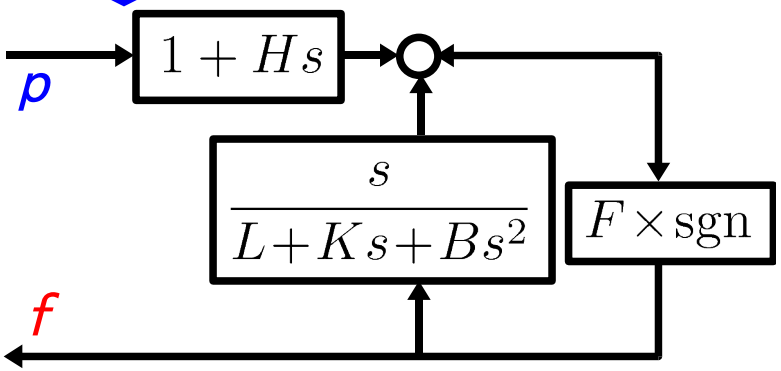
  - F：力の上限値．H：時定数.
  - 目標位置に時定数Hで指数収束.
- 切り替えが無限に高速であれば，力が飽和しない限り拘束は完全.
- しかし離散時間では切替によるチャタリングが生ずる.
- 通常は境界層を設けて，切り替えを滑らかにする.

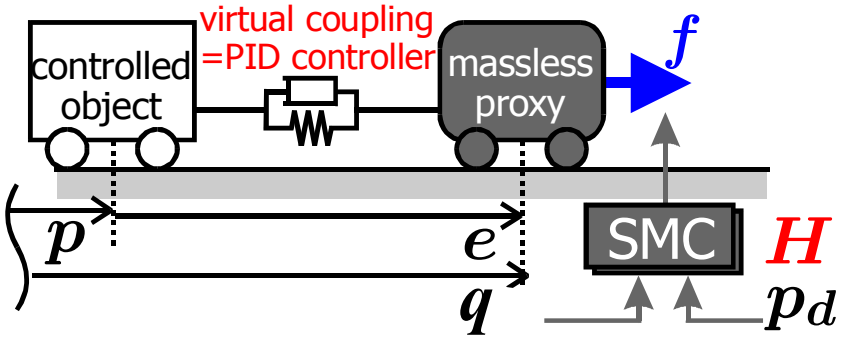
塑性のインピーダンス型離散時間表現はスライディングモード制御に拡張できる



- 質量のない仮想物体「プロクシ」にスライディングモード制御を施す.
- 「プロクシ」と制御対象はPID制御で接続

拡張(一般化)





- 「プロクシベースト・スライディングモード制御」(PSMC)

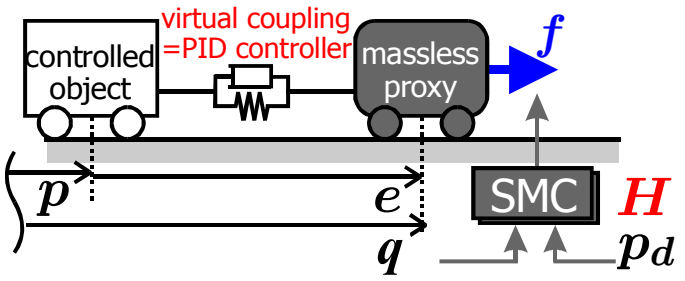
PSMCの制御則

[Kikuuwe & Fujimoto, 2006]

- 連続時間表現（代数微分方程式）

$$f = Ke + B\dot{e} + L\int e dt \quad (e = q - p)$$

$$f = F\mathrm{sgn}(p_d - q + H(\dot{p}_d - \dot{q}))$$



- 離散時間表現（代数方程式）

$$f_k = La_k + K\nabla a_k/T + B\nabla^2 a_k/T^2$$

$$f_k = F\mathrm{sgn}(p_{d,k} - q_k + H\nabla(p_{d,k} - q_k)/T) \qquad \left(a = \int e dt\right)$$

- 離散時間表現（**制御則**＝上式を解く計算手順）

$$\sigma_k := p_{d,k} - p_k + H(\nabla p_{d,k}/T - \nabla p_k/T)$$

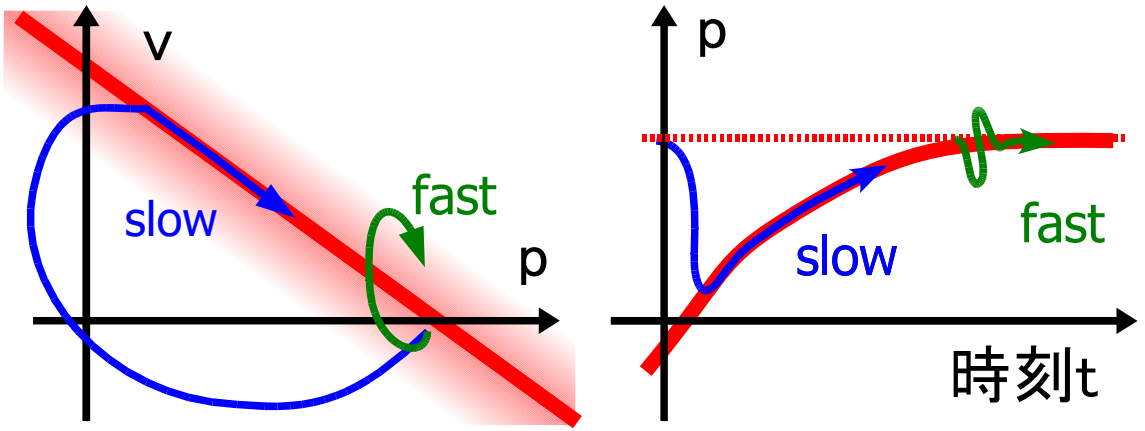
$$f_k^* := \frac{B+KT+LT^2}{H+T}\sigma_k + \frac{KH-B+LT(2H+T)}{(H+T)T}a_{k-1} - \frac{KH-B+LTH}{(H+T)T}a_{k-2}$$

$$f_k := F\mathrm{sat}(f_k^*/F)$$

$$a_k := \frac{(2B+KT)a_{k-1} - Ba_{k-2} + T^2 f_k}{B+KT+LT^2}.$$

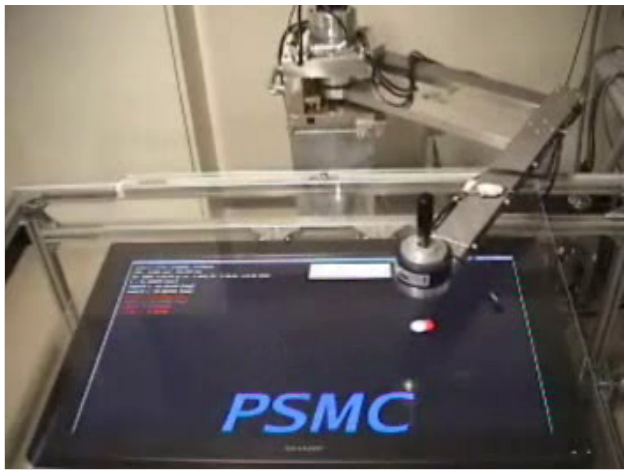
どんなメリットがあるの？

- PSMCは，PID制御と同等の高い剛性と，遅い応答を両立できる．⇒ PID制御と同等に正確で，より安全．
  - 局所的にはPID制御的な応答が支配的：
    - 小さい位置誤差からは素早く復帰．
  - 広域的にはスライディングモード制御が支配的：
    - 大きい位置誤差からは緩やかに復帰．一次の指数収束でオーバーシュート無し．
    - PID制御で同様の遅い特性を作ろうとすると，速度計測のノイズの影響を受けてしまう．

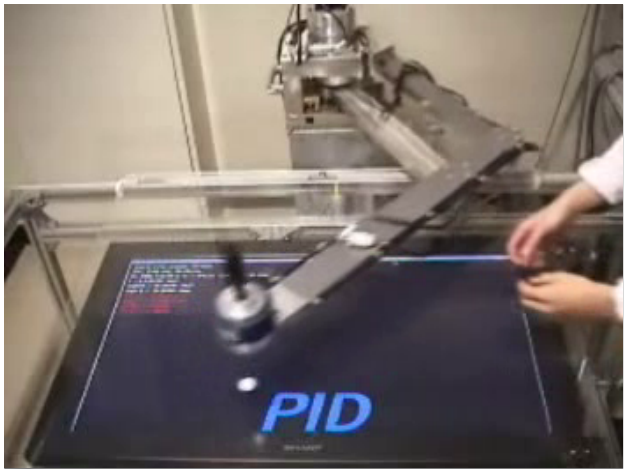


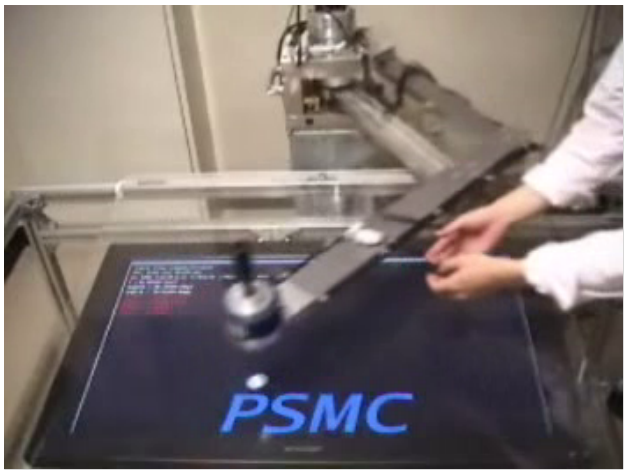
PID制御とPSMCの比較

- ステップ状の目標値入力に対する応答



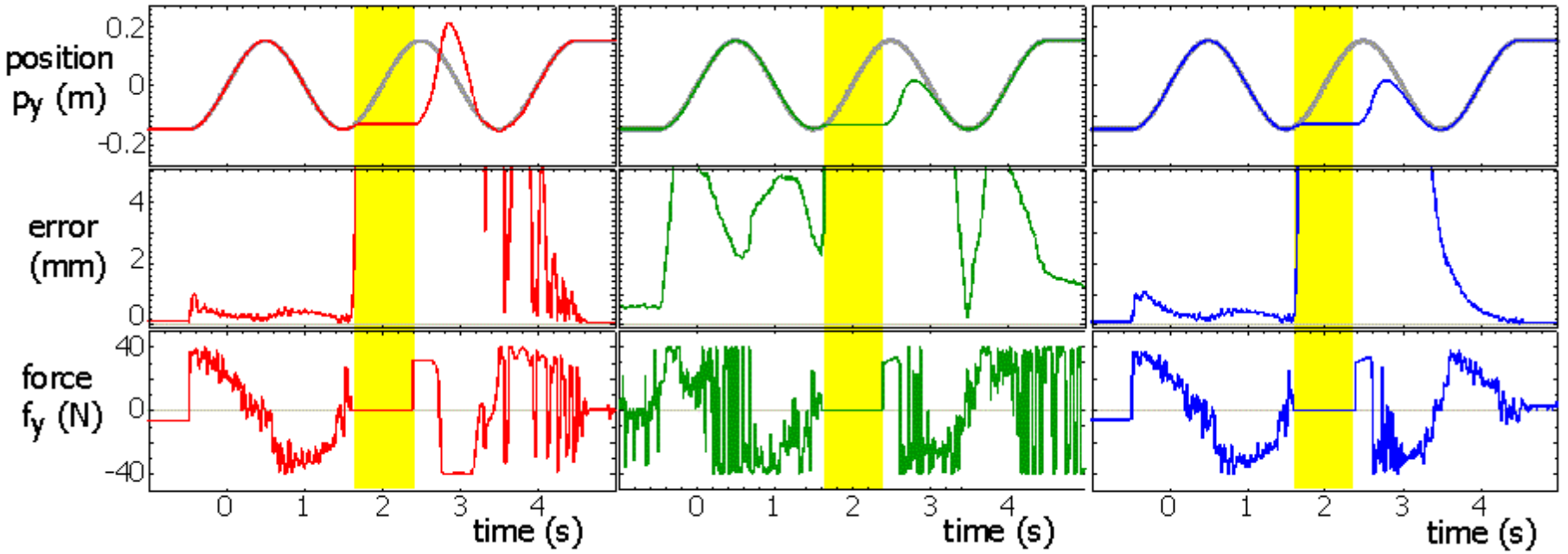
- 大きい外乱に対する応答





8の字運動：PD制御とPSMC

- アクチュエータ力を一時強制的にオフに



- **赤**：PD制御で微分ゲイン小：通常動作時は正確だが，以上発生時に安全ではない．．
- **緑**：PD制御で微分ゲイン大：アクチュエータ力が振動的になり，追従誤差が大きい．
- **青**：PSMC．通常動作時は正確だが，以上発生時にも安全．

導入事例

[Van Damme et al., 2007]
@ Vrije Univ. Brussel

2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Proxy-Based Sliding Mode Control of a Manipulator Actuated by Pleated Pneumatic Artificial Muscles

M. Van Damme, B. Vanderborght, R. Van Ham, B. Verrelst, F. Daerden, D. Lefeber
Robotics and Multibody Mechanics Research Group
Department of Mechanical Engineering
Vrije Universiteit Brussel
Pleinlaan 2, 1050 Brussel
michael.vandamme@vub.ac.be

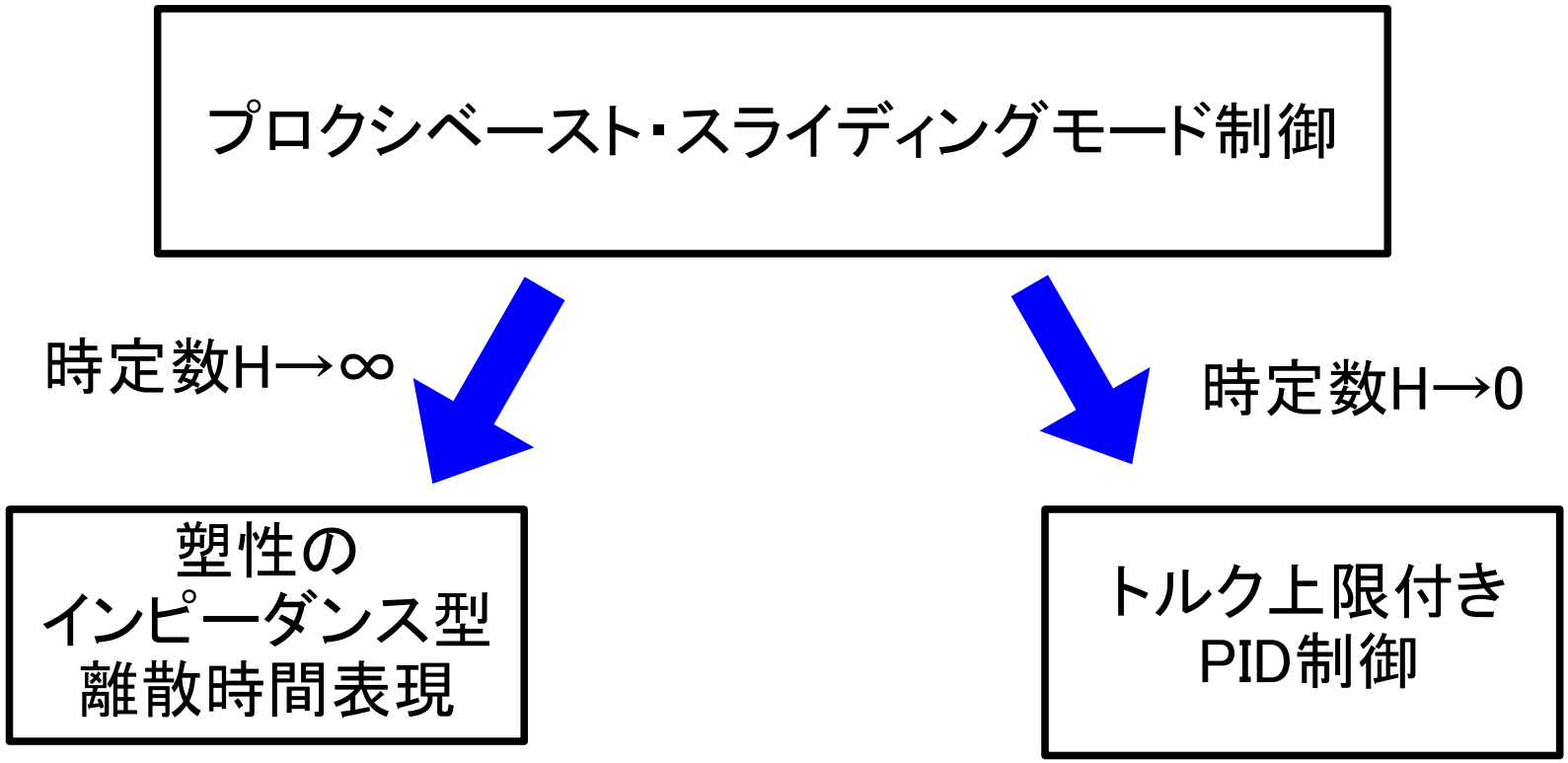
Abstract— Recently, Kikuuwe and Fujimoto have introduced Proxy-Based Sliding Mode Control. It combines responsive and accurate tracking during normal operation with smooth, slow recovery from large position errors that can sometimes occur after abnormal events. The method can be seen as an extension to both conventional PID control and sliding mode control.

In this paper, Proxy-Based Sliding Mode Control is used to control a 2-DOF planar manipulator actuated by Pleated Pneumatic Artificial Muscles (PPAMs). The principal advantage of this control method is increased safety for people interacting with the manipulator.

Two different forms of Proxy-Based Sliding Mode Control were implemented on the system, and their performance was experimentally evaluated. Both forms performed very well with respect to safety. Good tracking was also obtained, especially

Fig. 1. The manipulator. The series arrangement of Pleated Pneumatic Artificial Muscles is clearly visible.

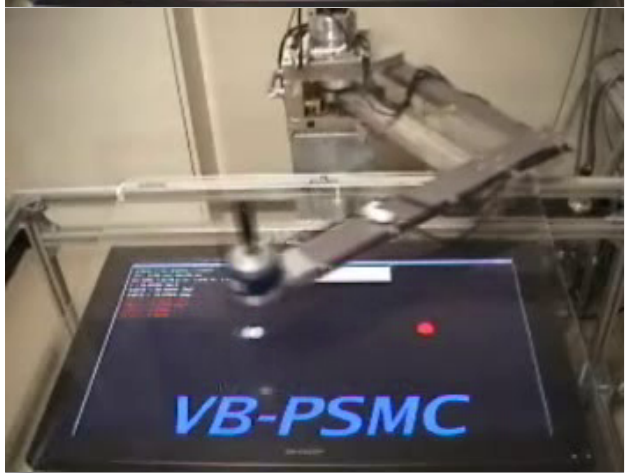
PSMC・PID制御・塑性モデル

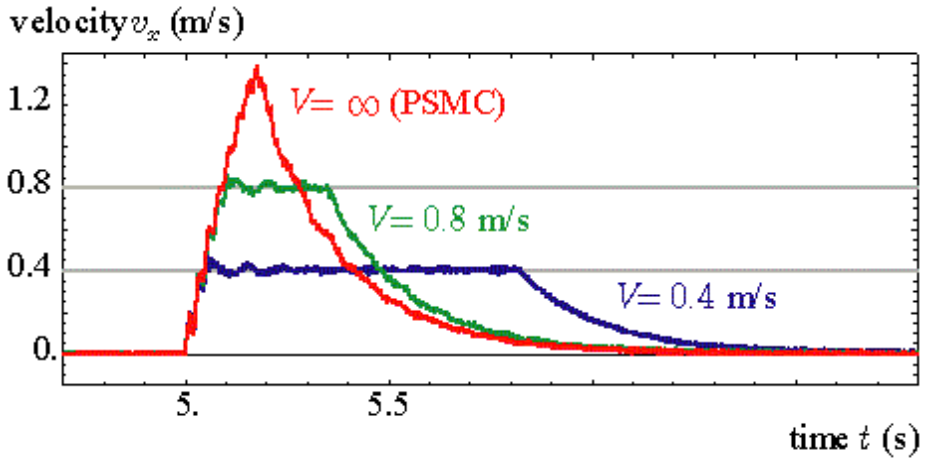


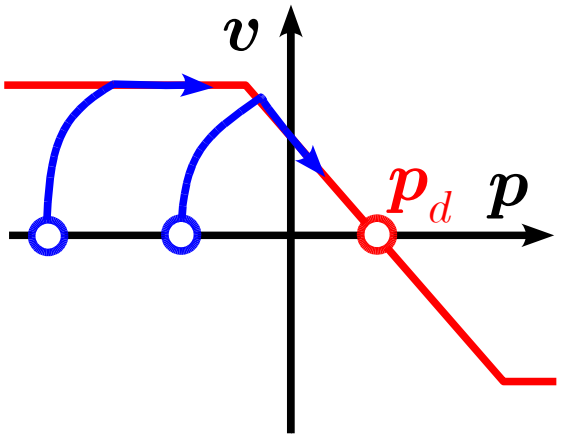
発展：速度制限つきPSMC

[Kikuuwe et al , 2006]

- 拘束多様体を適切に設計することにより，速度を制限
  ⇒ 誤った位置指令が与えられても，速度を制限







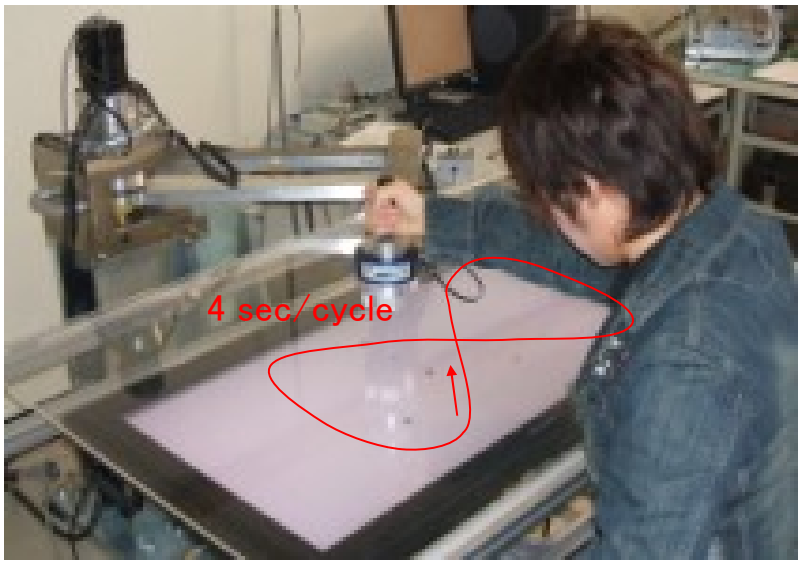
人と親和性(?)のあるロボット制御

安全だけではない？

- PSMC（時定数0.1秒）で8の字運動
- エンドエフェクタに手を添えて一緒に運動してみる.

  ⇒なぜか「しっくり」くる感じ.

  なぜ？

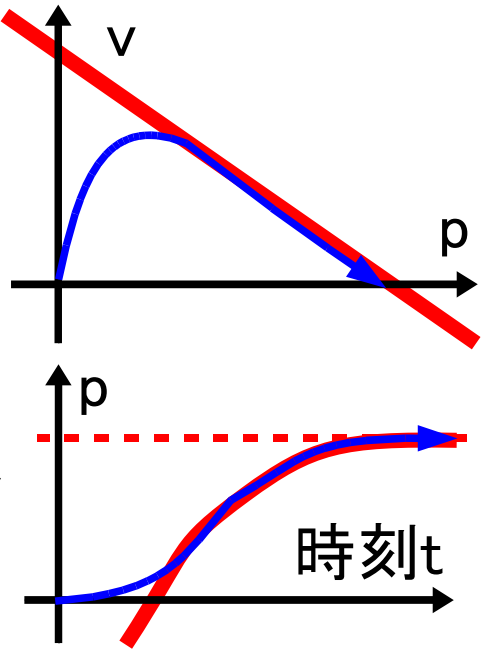


- アクチュエータの力の上限値を関節摩擦力以下にしてみる.

⇒ 静止摩擦のため，手で力を加えないとエンドエフェクタは動かない．わずかに目標軌道に引っ張られる感じがするのみ.

⇒ それでも運動してみると「しっくり」くる感じ．**なぜ？**

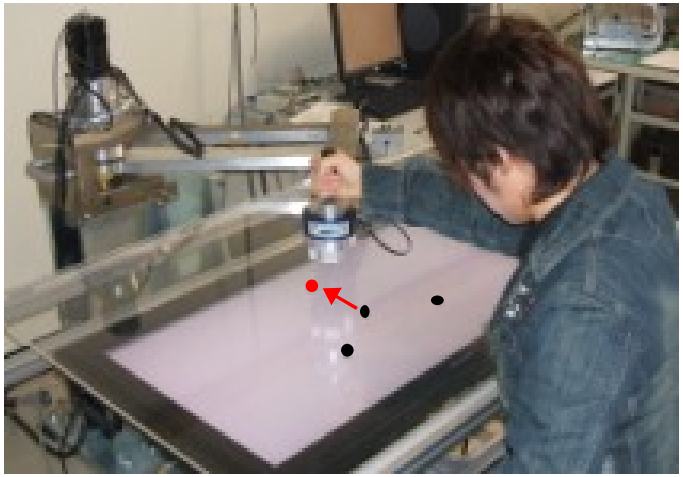
なぜ「しっくり」したのか？

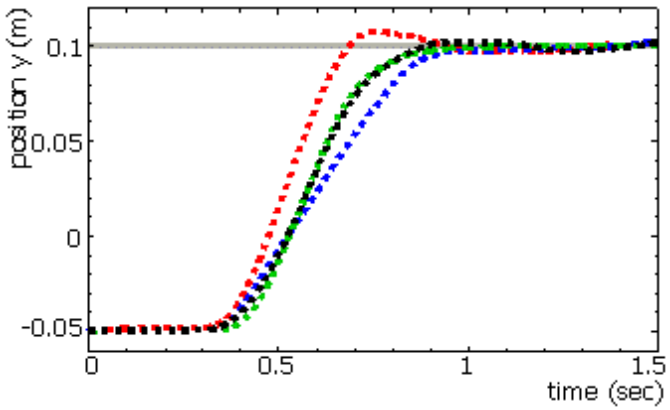


- 仮説：人間の到達運動は，ある時定数の指数関数で近似できる＝ 状態空間内である傾きの直線上に乗っている．
- PSMCで制御されるロボットの力は，状態空間内の直線を境界として切り替えられる．
- 「人間の時定数＞PSMCの時定数」であれば
  - ロボットは人間を常に目標位置方向に引っ張る．
  - ＝人間が減速しようとしていても，引っ張る
  - ⇒ オーバーシュート
- 「人間の時定数＜PSMCの時定数」であれば
  - ロボットは人間を減速する方向の力を出す．
  - ⇒ 余分な負担

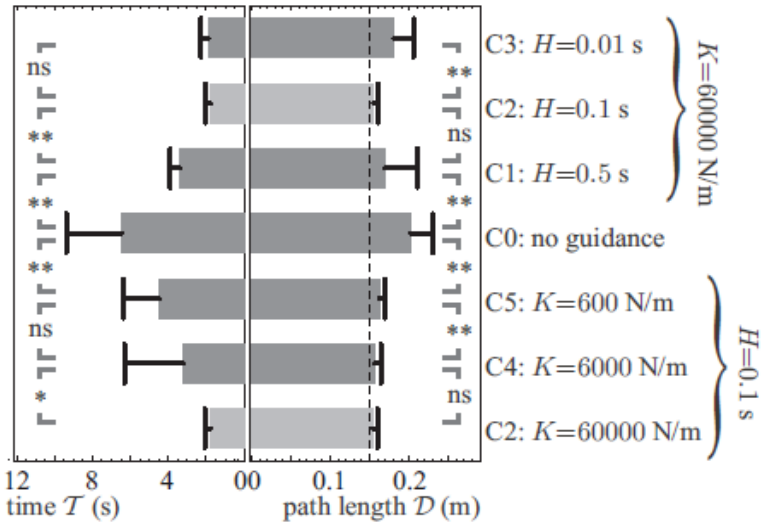
⇒「人間の時定数＝PSMCの時定数」が望ましい？

位置決め実験





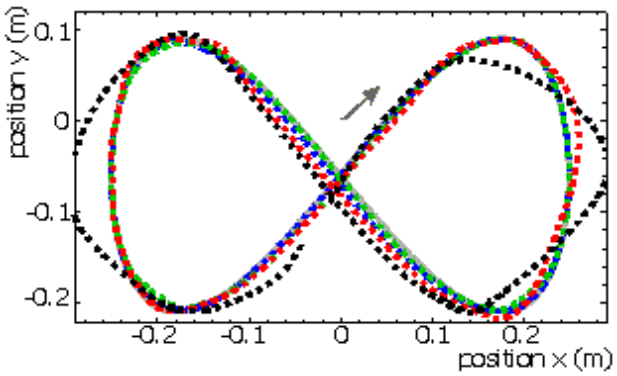
- 被験者は指定された位置にできるだけ速くハンドルを移動させる.
- ロボットはPSMCで制御．アクチュエータ力は静止摩擦力以下.

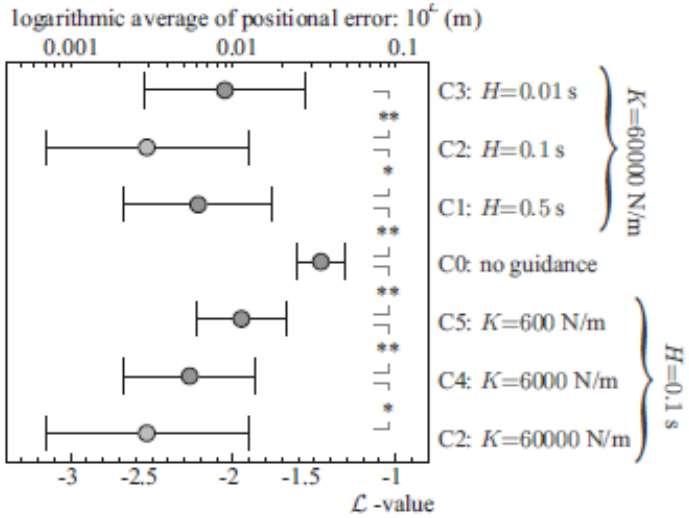


- 比例ゲイン（剛性）が高いほど，所要時間も移動距離も小さい
- 時定数が0.01秒：移動距離が長い（オーバーシュート）
- 時定数が0.1秒：所要時間も移動距離も最小
- 時定数が0.5秒：所要時間が長い.

軌道追従実験

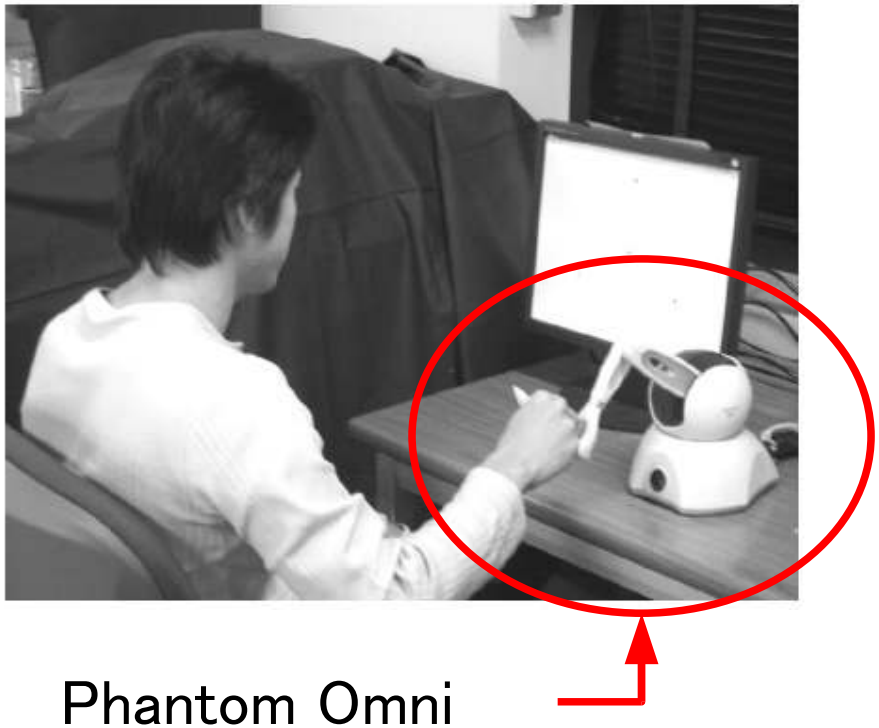
- 被験者は8の字の軌道にできるだけ正確にハンドルを追従させる.
- ロボットはPSMCで制御. アクチュエータ力は静止摩擦力以下.



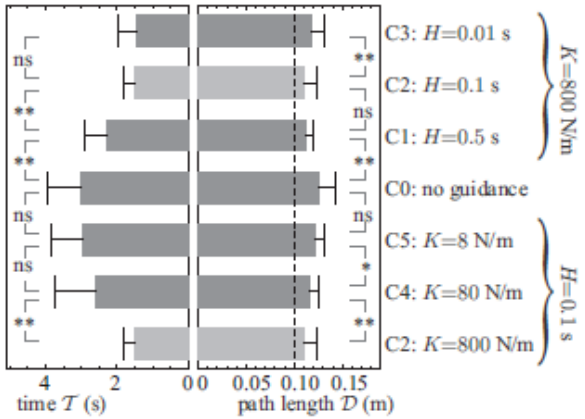


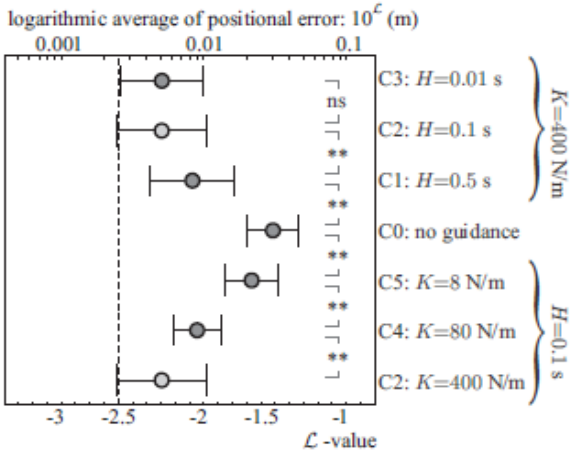
- 比例ゲイン(剛性)は高いほどよい.
- 時定数が0.01秒ではオーバーシュートが発生.
- 時定数が0.1秒で誤差平均が最小
- 時定数が0.5秒:誤差からの復帰が遅すぎる.

軽いロボットでも同様の実験



Phantom Omni





- 同様に，時定数0.1秒においてパフォーマンス（所要時間・移動距離・平均誤差）が高くなった.

この実験で得られたもの：

IEEE Trans. on SMC Part A で現在再査読

IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS, PART A: SYSTEMS AND HUMANS, VOL. X, NO. XX, XXXX 2007

A Guideline for Low-Force Robotic Guidance for Enhancing Human Performance of Positioning and Trajectory Tracking: It Should Be Stiff and Appropriately Slow

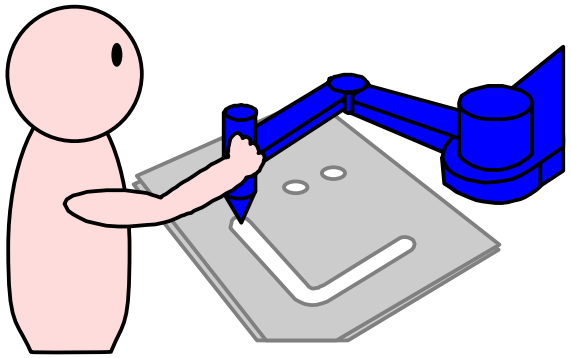
Ryo Kikuuwe, *Member, IEEE*, Takahiro Yamamoto, and Hideo Fujimoto

- 人間の位置決め・軌道追従のパフォーマンスをアップするための低出力ロボットガイダンスのためのガイドライン：それは硬く，そして適度にゆっくりするべき．
  （時定数で0.1秒程度）
- 「硬くて適度にゆっくり」な応答特性は，PIDでは不可能．PSMCでなければ実現できない．

何の役に立つの？

たとえば，

- 製造産業の現場：決められた軌道に沿った塗装・溶接など
  - 技術的には全自動化が可能でも，安全性や信頼性の面で人間の介入が随時必要な場合．
- リハビリ（自動介助運動，active assistive motion）



⇒人間・機械協調系の1つのクラス：
"Low-force kinesthetic guidance"

- 作業者は決められた目標位置（目標軌道）にツールを位置決めする（追従させる）．
- ロボットは，ツール位置を誘導するために小さい力を出す．
- ロボットの制御器は目標軌道を予め持っている．
- 作業者は目標位置を知っており，視覚的にもそれを認識できる．
- **ロボットの力は十分に小さく，作業者が意図すれば容易に目標軌道から逸脱することができる．**

この研究が示唆すること（希望的観測）

- **「0.1秒」**はおそらく人間の運動と人間・機械間の相互作用を特徴付ける「マジックナンバー」.
  - 人にとっての使いやすさ，使い心地に関連？
    - 使いやすい引き出し，操作レバー，スイッチ，・・
  - （粘性／剛性）or（粘性／慣性）は 0.1秒程度がよい？
  - 人との協調に適した滑らかさ・遅さの指標？

- **人間に近い動きをするロボット ＝ 人間との協調作業に適したロボット？**
  - 人間の随意運動を記述する微分方程式が見つかれば，人の運動に近い滑らかな動きを関節摩擦の大きいロボットで実現できるかもしれない.

まとめ

- 離散時間で「塑性」を扱う方法を紹介した.
- 安全な汎用位置制御手法『プロクシベースト・スライディングモード制御(PSMC)』を紹介した.
- 『Low-Force Kinesthetic Guidance』および人間とロボットの力学的相互作用に関する実験結果と展望を示した.

今後の課題(現在のテーマ)

- PSMCの拡張

⇒ 比較的低コストな動力機構(サーボモータ・ギア減速機など)を用いて，人間と協調・共存できるロボットを実現したい.

- 人間とロボットの協調作業・協調運動・力学的相互作用に必要なエッセンス(?)を見つけたい.