

July 16, 2020

はじめに
現在の状況
おわりに

なぞり運動における内部モデルと 習熟メカニズムの模倣と応用

1715038 清水 豪士

富山県立大学 情報基盤工学講座

July 16, 2020

要旨

情報数理工学と制御工学を融合した基盤研究にもとづいて、ターゲットトラッキングタスクでなぞり運動における内部モデルのモデル化と習熟メカニズムを解明する．拡張カルマンフィルタと報酬駆動システムの枠組みで自律分散制御の基盤技術を開発する

PsychoPy というツールを使い進めていく

拡張カルマンフィルタ

カルマンフィルタとは状態空間モデルにおいて、内部の見えない状態を効率的に推定するための計算手法拡張カルマンフィルタとは、カルマンフィルタを非線形システムに対応するように拡張したもの

報酬駆動システム

意思決定の主体が戦略を選び行動した結果、報酬を原動力として駆動しているとみなすことができ、ペイオフにもとづく効用関数の変化を報酬として入力しているもの

似たような研究として、手の動きと視線の動きの関係性について研究したものは多数見られたが、「なぞり運動」に重きを置き、なぞるべき物と実際になぞってみたものとの誤差についての研究は少なかった。

PC を使って心理学実験を行うためのツールとして PsychoPy がある.

刺激画像の表示時間の指定をしたり、刺激画像が表示されてからのボタンを押すまでの反応時間を記録するといったことができる

PsychoPy は、Python というプログラミング言語を用いて PC に指示を出す

PsychoPy には Corder と Builder という機能がある

Builder

自分でプログラムを書くのではなく、アイコンを配置して実験を作成する

作成した実験は Python のスクリプトに変換してくれる

アイコンには刺激という項目で画像表示、文字の表示や音の再生などがあり、反応という項目でマウスのクリックやキーボードを押すなどがある

Corder

直接コードを書くことができる

今回の実験の作成は Builder を使っている

はじめに
現在の状況
おわりに

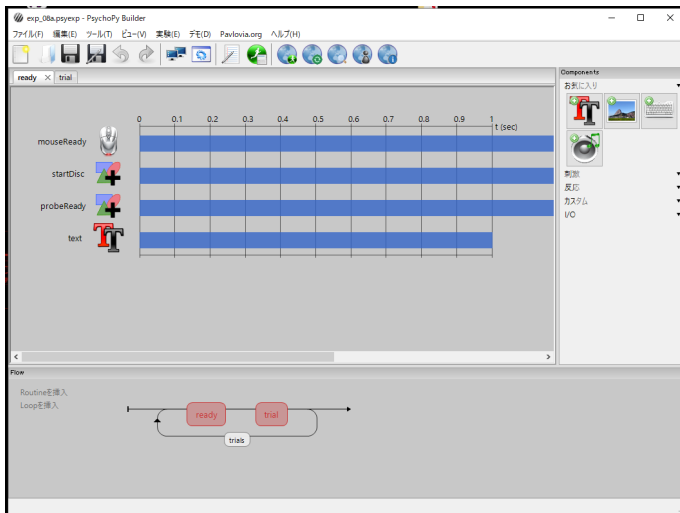


図 1 : Builder

はじめに
現在の状況
おわりに

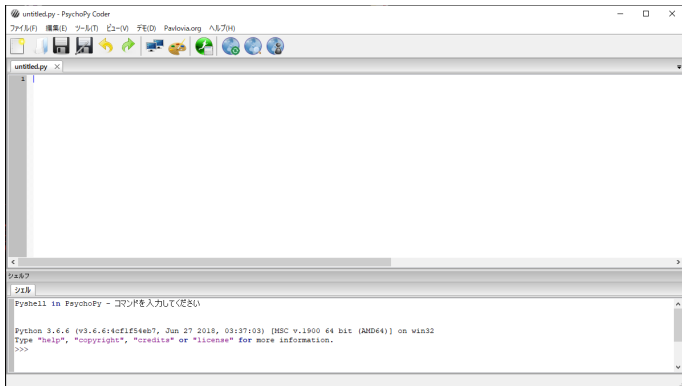


図 2 : Corder

はじめに
現在の状況
おわりに

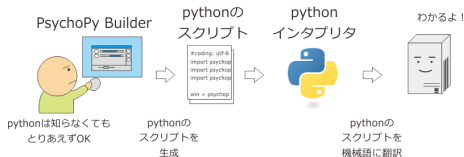
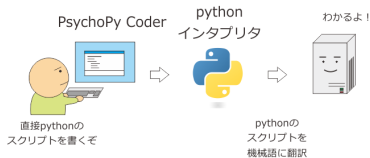


図 3 : Builder と Coder のイメージ

実験概要

鏡映描写課題と類似の課題をP Cで実現させる

鏡映描写課題

鏡映描写課題とは、鏡に映った自分の手の像を見ながら図形をペンでなぞる課題

鏡を見ながら描画するため、前後方向に手を動かしたときに視覚像の動きが逆転しているため、うまく図形をなぞるのが難しい
しかし、何度も練習を繰り返すことで、次第に手と鏡像の動きの関係が学習され素早く間違わずになぞることができる

実験は星の半分の形をなぞるものになっている。

全部で 10 回の試行をするようになっている。(ランダムに出現)

試行の 1 回 1 回で始点と終点が変化するようになっている。

マウスを動かしたら左右は同じように動くが、上下は反転している

この実験ではなぞるべき線をきちんとなぞれているか、いないかを判別し、エクセルに結果を出力する。

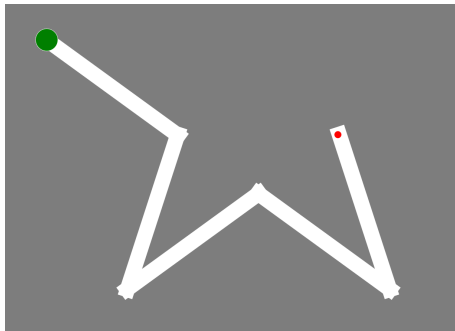


図 4：実験最中

赤点が自分が動かしている対象のものを示している
緑点が目的地（終点）を示している
赤点が緑点に到達したら、次の試行に切り替わるようになっている
きちんとなぞれている時を True、なぞれていない時を False といった形で示す

はじめに
現在の状況
おわりに

	62	142		False
	62	142		False
	62	142		False
	62	142		False
	62	142		False
	62	142		False
	62	142		False
	62	142		False
	62	142		False
	62	142		False
	62	142		False
	62	142		False
	62	142		False

図 5：実験結果の一部

これは結果を並び替えている

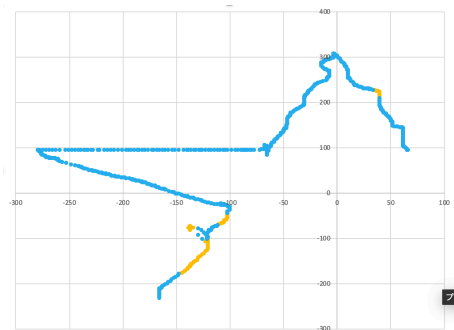


図 6：実験結果をエクセルでグラフ表示

図 5 は 10 回の試行のうちの 1 つをエクセルでグラフ化したものである。

青点はなぞるべき線をなぞった (True) のを示している、オレンジ点はなぞれなかった (False) のを示している。

このグラフは約 1200 個のデータで構成されている。

前回の研究会で挙げられたもの

- 1：四角の4隅にチェックポイントを設ける
- 2：始点、終点を一緒に座標に配置する
- 3：実際にデータを取得し、習熟しているのかを測る

1、 2 について

16/23

今回はこれに多くの時間がかかってしまった。

いろいろな方法をやって見た結果、やっていてわかりやすいように、各ポイントを通過するごとに色を変化するようにした

この時に最初から2つ目のポイントに行っても色が変化しないよう、ちゃんと1つ目のポイントを通ってから2つ目のポイントを通過しないと色が変化しないようになっている

そして、左上、左下、右下の3つのポイントの色が変化した状態でゴールに到達しないと、次の試行に移らないように設定した

1、2について

17/23

はじめに
現在の状況
おわりに

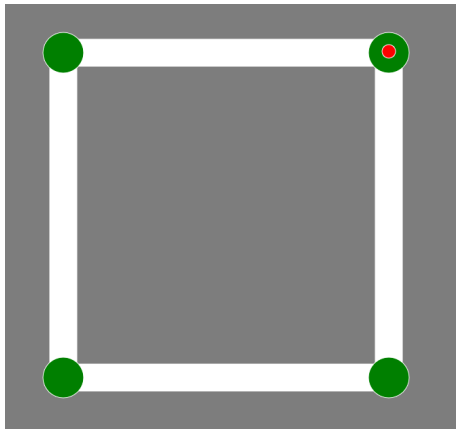


図 7： スタート時

1、2について

18/23

はじめに
現在の状況
おわりに

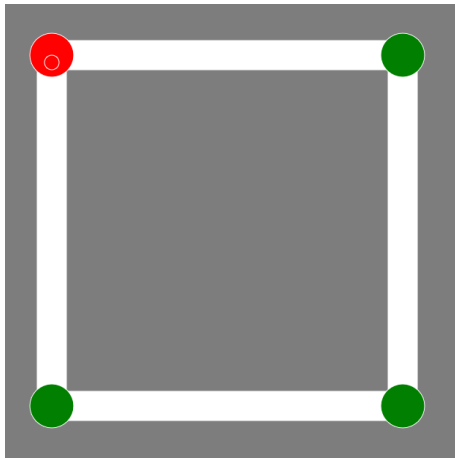


図8：1つ目のポイント通過

1、2について

19/23

はじめに
現在の状況
おわりに

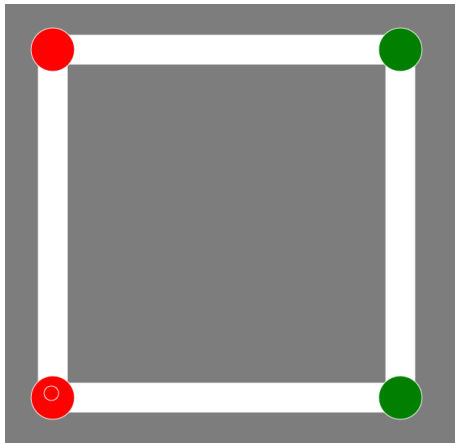


図 9：2つ目のポイント通過

1、2について

20/23

はじめに
現在の状況
おわりに

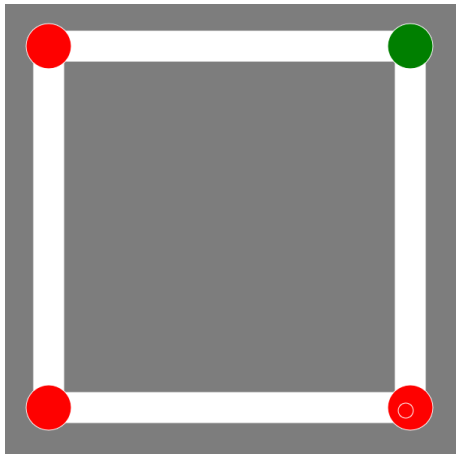


図 10： 3つ目のポイント通過

1、2について

21/23

はじめに
現在の状況
おわりに

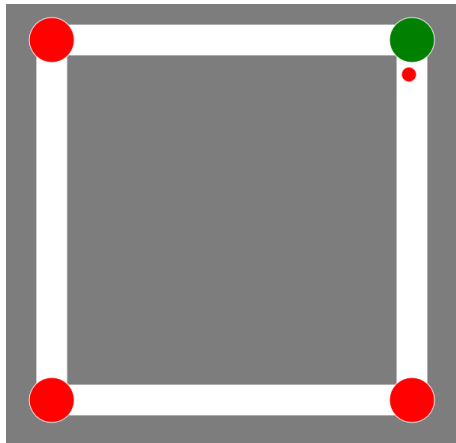


図 11：ゴール手前

今回は1、2に時間がかかってしまい、3のデータ取得と習熟の確認はできていない

いまのところは実験の条件で追加するものがないので、次からはデータ取得ができると考えている

これに関しては、まず自分で実験をやりデータ取得をし、実験に問題がないことを確認できたら研究室の人たちに協力してもらい違うデータを取得しようと考えている

今回は実験が完成した

次からは実際にデータを取得し、習熟の確認を行う
また、理論の方からのアプローチをしていく予定である