

# 2013 年度エンジニアリング・プラクティス夏課題

## プログラミング課題

作成：菅野研 M1

次項から始まる[必修課題]と[選択課題 3 問のうち 1 問以上]に取り組んでください。つまり、必修課題 1 つと選択課題 1 つの合計 2 つの課題は必ず提出してください。余力のある人は、是非他の選択課題にも挑戦してみてください。その場合は加点します。

【提出期限】 **10月1日（火）18:00 まで**（後期エンブラの初回です）。

【提出形式】課題毎にソースコード、プログラムの動作に関するファイル（csv ファイルなど）を一つのフォルダにまとめ、zip に圧縮してメールに添付して提出。

【フォーマット】下記参照（**厳守!**）

---

フォルダ名：エンブラ夏課題\_P\*\_提出者氏名

P: プログラム課題であることを示す添字

\*: 課題番号（必修課題であれば 1 が入る）

メールタイトル：2013MMDD\_エンブラ夏課題\_P\*\_提出者氏名

MMDD: 提出する日付。

例. 選択課題番号 2 を提出する場合

メールタイトル：20130716\_エンブラ夏課題\_P2\_佐藤高志

添付ファイル：エンブラ夏課題\_P2\_佐藤高志.zip

---

【提出先】 [m1-ml@sugano.mech.waseda.ac.jp](mailto:m1-ml@sugano.mech.waseda.ac.jp)（質問もここまで）

### 【注意事項】

1. フォーマットは必ず守ること。
2. 動作がわかるようにコメントをこまめに入れる、内容が煩雑な部分は関数化する、ファイルを分けるなど、他人に見やすいソースを作るように心がけてください。それ自体も評価対象となります。初回近くで配布した「エンブラ関係プログラミングルール」も熟読すること。
3. コンパイルが通らないような明らかに動作を確認していないソースコードや、所定の動作をしないプログラムを提出しないでください。どうしてもできない、分からないなどがあれば、“**早め**に” 質問するようにしてください。可能な限り対応します。
4. ソースを Web、他人のものからコピーしてこないでください。判明した場合、**評価しません**。同様に、明らかにやる気が認められないもの、雑なものは著しく評価が低くなります。

## < 1. 必修課題：アームロボット制御 >

Fig.1 に制御対象のアームロボットを示す. このロボットは全方向台車の上に 2 自由度リンク機構のロボットアームを備えたロボットである. 台車は  $x$  方向,  $y$  方向に移動でき, その場で  $z$  軸回転を行うことができる.

このロボットが Fig.2 に示すような 3 次元空間内にあるとする. ロボットは格子上の黒点間を移動し, 黒点の上でのみ停止, リーチング,  $z$  軸回転動作が行えるものとする.

この条件において, 下記のタスクを実行するようなシミュレータを構築して欲しい.

- ① 空間内の任意の場所に対象物を設置する. キーボード入力, もしくは外部ファイルを読み込んで対象物の個数と各対象物の三次元座標 ( $x,y,z$ ) を取得する. このとき, 対象物を設置するのは黒点の上でなくてもよい.
- ② ロボットは部屋に設置されたセンサにより, 設置された対象物の絶対位置を正確に計測できるものとする.
- ③ 各対象物についてリーチングできるかを判定し, 出来ない場合は把持対象から除外する.
- ④ 把持出来る対象物が判定できたら, 一つずつの対象物に対して台車を駆動して, リーチングできる点へ移動し, リーチング動作を行う. ロボットは初期条件として, Fig.2 に示すホームポジションに  $x$  方向を向いて停止しているものとする.
- ⑤ 移動の履歴と関節角度は全て CSV に記録し, どのような動作をしたかを確認できるようにする.
- ⑥ 全ての対象物を把持できたらホームポジションに戻って動作終了.

### 【注意点】

- 授業で作ったソースコードを十分に活用しましょう.
- 空間の絶対座標系とロボット中心座標系との違いに注意しましょう.
- 台車の移動はロボット中心座標系で表記してください. ロボットから見て斜めに進む場合は  $x=\bigcirc$ ,  $y=\square$  というふうに表記する必要があります. 個人的には移動する際は  $x$  軸方向,  $y$  軸方向いずれか単体のみにしたほうが扱いやすいかと思います (つまり, 体の正面の方向に移動するか, 真横の方向に移動するか).
- リーチングを行う順番, 台車の軌道は自由に決定して構いませんが, どのような方針で決定したかをコメントなどで表記しておいてください.

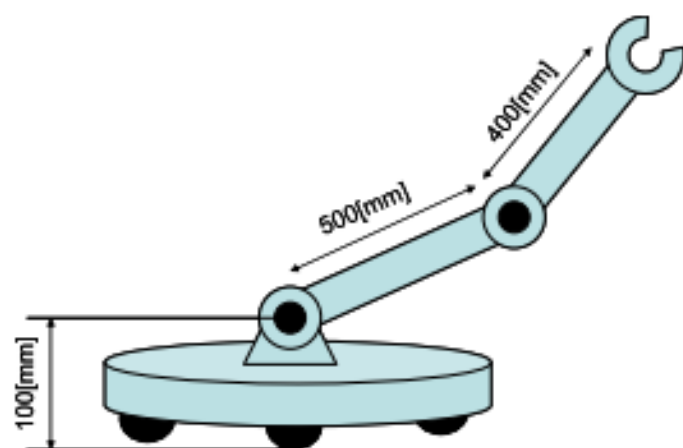


Fig.1 ロボット機構図

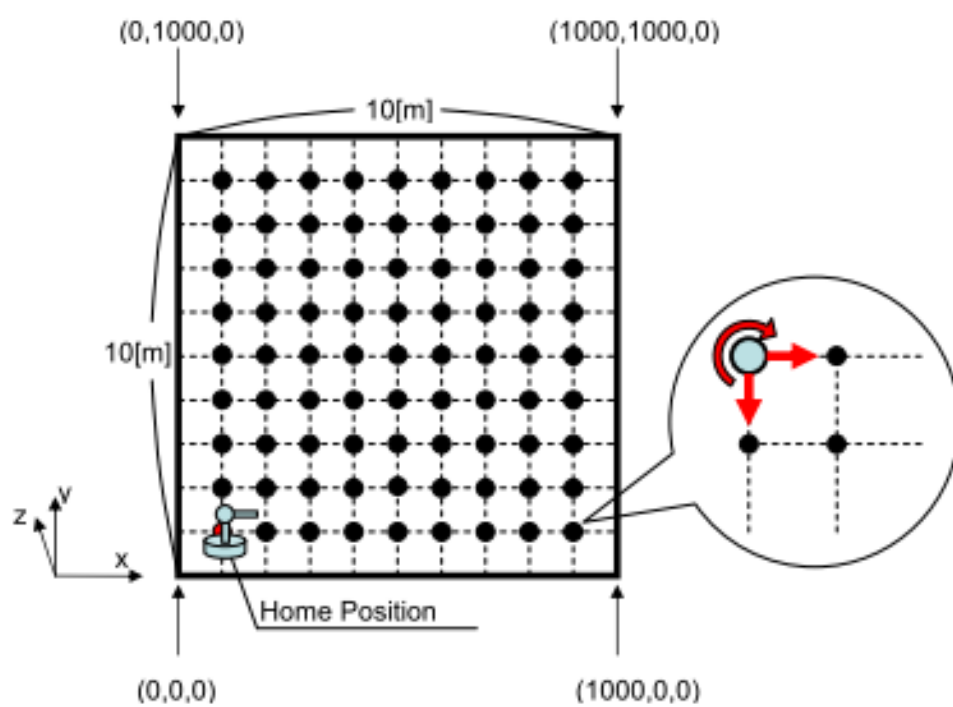


Fig.2 フィールド図

## &lt; 2. 選択課題：自動販売機プログラム &gt;

以下のような動作をする自動販売機シミュレーションのプログラムを作成してください。例えば、以下のような実行結果を出力できるようにしましょう。

```
==== メニュー =====
1. お茶 ..... 150 円
2. コーヒー . 140 円
3. 牛乳 ..... 95 円
商品を選んでください[1-3]: 0

==== メニュー =====
1. お茶 ..... 150 円
2. コーヒー . 140 円
3. 牛乳 ..... 95 円
商品を選んでください[1-3]: 2
あと 120 円足りません
お金を投入してください: -1
あと 120 円足りません
お金を投入してください: 5
10 円以下は投入しないでください
あと 120 円足りません
お金を投入してください: 15
10 円以下は投入しないでください
あと 120 円足りません
お金を投入してください: 1000
おつり
( 100) x 8 枚
( 50) x 1 枚
( 10) x 3 枚
ご利用ありがとうございました
```

以下の機能は必須機能とします

✓ メニューの選択の祭 1,2,3 のいずれかが入力されるまで選択を続ける

- ✓ 金額の入力では通常のお金の値の入力以外は受け付けないこと．使える値は以下のようである：
  - 10 円、50 円、100 円、500 円、1,000 円、5,000 円、10,000 円
- ✓ 金額の入力では商品の金額以上が入力されるまでは繰り返し入力させる
- ✓ おつりは、各紙幣、硬貨ごとの枚数を表示する．0 枚の場合は表示しない
- ✓ 商品の選択では、ある「管理パスワード」を入力すると「管理者メニュー」が表示される．このメニューでは、管理者が以下の項目が出来ることとする
  - 在庫の確認

===== 商品の在庫 =====

お茶 ..... 10 個

コーヒー..... 7 個

牛乳 ..... 2 個

- 在庫の追加・削減
- 紙幣と硬貨の数の確認

===== 紙幣と硬貨の枚数 =====

10 円..... 50 枚      1000 円..... 48 枚

50 円..... 35 枚      5000 円..... 5 枚

100 円..... 24 枚      10000 円..... 2 枚

500 円..... 49 枚

- 紙幣と硬貨の追加・削減
- \* プログラムを初めて起動する祭、管理者が商品の在庫と紙幣・硬貨の枚数を入力してから販売を開始する
- \* つまり、毎回の販売では在庫がだんだん減って、紙幣・硬貨の枚数も変わっていく
- ✓ このプログラミングは自動販売機のシミュレーションなので、実際の自動販売機の動作に出来る限り似せてください．

### < 3. 選択課題：組み合わせの計算 >

あなたはエンブラの回路製作課題のために、「面倒くさいなあ」と思いつつ秋葉原と新宿へパーツを買いに行き、以下の 7 枚の領収書を貰って来ました。

番号	購入品名	金額 (円)	購入店
1	ユニバーサル基板	100	秋月電子通商
2	スピーカー	280	マルツパーツ館
3	ギアボックス	746	千石電商
4	アルミ板	578	東急ハンズ
5	抵抗とコンデンサ	63	千石電商
6	電池ボックス	70	秋月電子通商
7	OP アンプ	315	千石電商

今、これらの領収書を任意の枚数、組み合わせでエンブラ会計に提出することで、最大 1500 円まで返金してもらうことができます。できるだけ損しないように、**なるべく返金額が高くなる領収書の組み合わせ(枚数は任意)を導出するプログラム**を作成してください。

#### 【要点】

- 合計 1501 円以上となるような組み合わせで領収書を提出してしまうと 1 円も返金されません。
- 全ての組み合わせをただ闇雲に調べるプログラムは評価が低いです。
  - 組み合わせの総数は、それぞれの領収書を選ぶ・選ばないで 0, 1 と考えれば、 $2^7 = 128$  通り。これくらいなら全部調べてもたいした時間はかかりませんが、領収書が 20 枚だったら  $2^{20} = 1048576$  通り、30 枚なら  $2^{30} = 1073741824$  通り、…、100 枚だったら？あんまり計算したくはありませんね。そういうことです。
- 領収書の購入品名、金額、購入店は、簡単に内容の変更や追加、削除ができるように外部の csv ファイルなどから読み込めるようにしてください。その際、領収書番号は外部ファイルに記述するのではなく、プログラムで読み込んだ時に自動的に付与するようにしてください。
- 計算結果は、外部ファイル（形式は自由ですが、それをそのまま同じプログラムで読み込めるようにしてください）に保存できるようにしてください。
- 実行したプログラム内において、インタラクティブ（対話的）に金額順や購入店準などでソート（並び替え）ができるようにしてください。また、ファイルに保存するときにそれを反映してください。
- 

#### 【動作例】

>csv ファイルを読み込みます...読み込み OK.

>領収書の内容を表示します.

\*\*\*\*\*

番号 購入品名 金額[円] 購入店

1	ユニバーサル基板	100	秋月電子通商
2	スピーカー	280	マルツパーツ館
3	ギアボックス	746	千石電商
4	アルミ板	578	東急ハンズ
5	抵抗とコンデンサ	63	千石電商
6	電池ボックス	70	秋月電子通商
7	OP アンプ	315	千石電商

合計金額 : 2152 [円]

\*\*\*\*\*

>返金額が高い組み合わせを計算中(目標金額 : 1500[円])...計算が終了しました.

>求めた領収書の組み合わせを表示します.

\*\*\*\*\*

番号 購入品名 金額[円] 購入店

3	ギアボックス	746	千石電商
4	アルミ板	578	東急ハンズ
5	抵抗とコンデンサ	63	千石電商
6	電池ボックス	70	秋月電子通商

これはあくまでも一例です.  
これが最適解だとは限りません.

合計金額 : 1457 [円]

\*\*\*\*\*

>コマンドを選択してください [1:ソート, 2:ファイルに保存, q:終了] : 1(Enter)

>何でソートしますか? [1:金額(昇順), 2:金額(降順), 3:購入店(昇順), 4:購入店(降順)] : 1(Enter)

\*\*\*\*\*

番号 購入品名 金額[円] 購入店

5	抵抗とコンデンサ	63	千石電商
6	電池ボックス	70	秋月電子通商
4	アルミ板	578	東急ハンズ
3	ギアボックス	746	千石電商

合計金額 : 1457 [円]

\*\*\*\*\*

>コマンドを選択してください [1:ソート, 2:ファイルに保存, q:終了] : 2(Enter)

>保存ファイル名を入力してください : ryousyusyo.csv

>[ryousyusyo.csv]を作成しました.

>コマンドを選択してください [1:ソート, 2:ファイルに保存, q:終了] : q(Enter)

>プログラムを終了します.

#### **Ex 課題（取り組んだ場合は加点）**

- 上記のプログラムを、読み込む外部ファイルの内容（金額、枚数など）、および目的とする最大返金額が変わっても対応ができるように改良してください。枚数がすごく多くても(100枚とか)計算できるのが望ましいです。その際、最適解を計算することが難しければ、近似解の計算でも構いません。

（さらに余力があれば）

- 計算アルゴリズムによる近似解の精度の違いや、計算時間の違いについて実際にプログラムを動かして調べ、グラフや表などを用いてレポートを作成せよ。



#### < 4. 選択課題：学習・ニューラルネットワークによる関数近似 >

多層のニューラルネットワーク (Feedforward Neural Network) [1]は、教示データに内在する規則性を獲得することが可能である。実際に、3 層のニューラルネットワークは隠れ層のニューロンが十分にあれば、結合重みを上手く選ぶ事により、任意の関数を近似できることが証明されている[2].

そこで、このニューラルネットワークを C 言語により実装し、適当な教示信号（ネットワークの出力層の値域に注意して用意する：例えば正弦波）を用いて学習させ、未学習のデータに関してもロバストな対応が出来るかどうかを試した例を示せ（汎化能力の確認）.

なお、学習に必要なデータはファイルから読み込み、プログラム中では少なくとも以下の構造体・関数を使用すること.

- ・ 構造体

- ニューラルネットワークに関する変数を有する構造体 `NeuralNetwork`

- ・ 関数

- 学習に必要な変数の初期化をする `InitParameters`
- 教示信号を変数に格納する `AddTarget`
- Forward Propagation（前向き計算）を行う `ForwardPropagation`
- エラーを算出する `GetError`
- Backward Propagation（後ろ向き計算）を行う `BackwardPropagation`
- 学習によってパラメータを更新する `UpdateParameters`

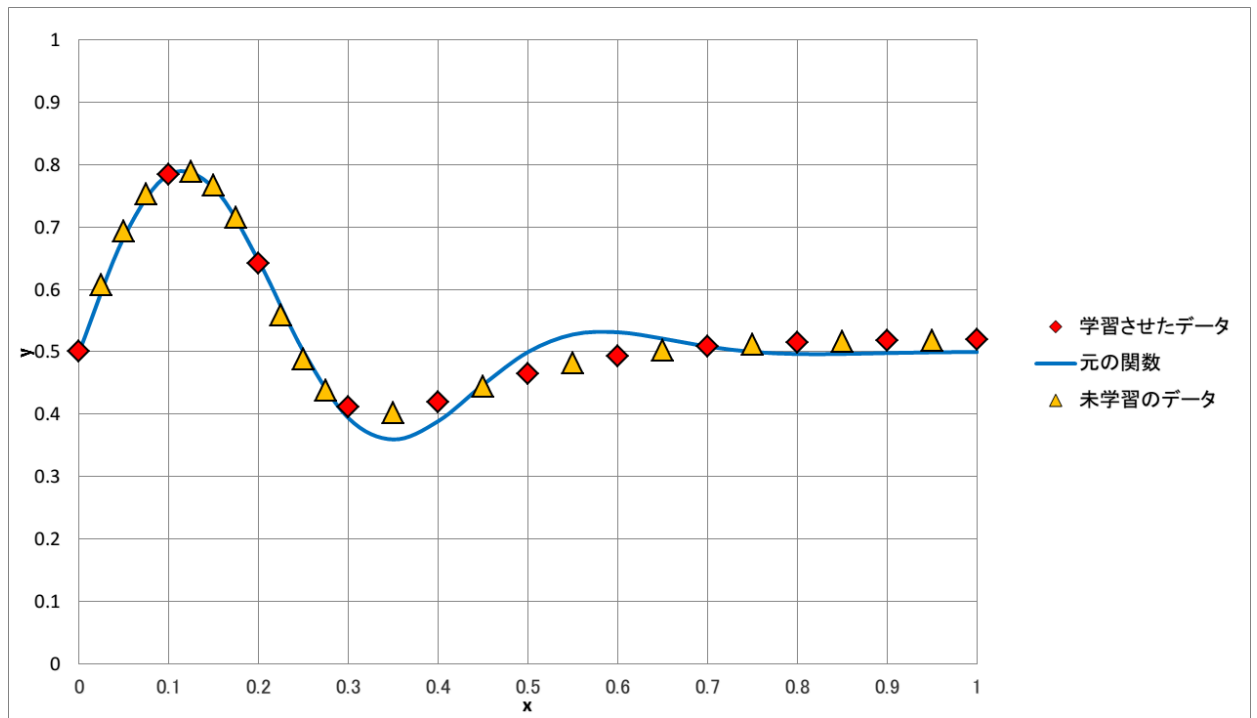
また、ニューラルネットワークは様々な場面で利用することができるため、以下に挙げるようなパラメータの設定に関して汎用性の高いプログラムにするとということを心がけること（プログラム自体を書き換える必要なく、設定ファイルの読み込みやコマンドラインを使用したインタラクティブな設定が可能であるようにする）.

- 入力、隠れ、出力層の次元
- 学習回数
- 学習に必要なパラメータ（学習率や減衰係数等）

ソースファイルと、自分の作成したプログラムが課題を達成出来ていると示すもの（グラフ等）を提出.

参考にした文献などがあれば、それもどこかに書いておいてください.

### 【提出するグラフの例：Gabor 関数の学習】



### 参考文献

- [1] David E. Rumelhart, Geoffrey E. Hinton, Ronald J. Williams. Learning representations by back-propagating errors, Nature, Vol.323-9, pp.533-536, 1986.
- [2] Ken-chi Funahashi, On the approximate realization of continuous mapping by neural network. Neural Networks, Vol.2, pp.183-192, 1989.