

# 平成27年度 知能機械情報学専攻

## 大学院修士課程入学試験問題

### 「知能機械情報学（科目）」

試験日時：平成26年8月18日（月）14：00～16：00

#### 注意事項

1. 試験開始の合図があるまで、この問題冊子を開かないこと。
2. 問題は3題出題されている。問題1（必答問題）は必ず解答し、問題2Aおよび問題2B（選択問題）から1題を選択して解答すること。
3. 問題の解釈に複数の可能性が考えられる場合は、適宜言葉の定義や条件などを付加して解答してよい。
4. 問題冊子に落丁、乱丁、あるいは印刷不鮮明な箇所があれば申し出ること。
5. 答案用紙は2枚配布される。枚数を確認し、過不足があれば申し出ること。問題ごとに1枚の答案用紙を用いて解答すること。解答を表面で書ききれない場合は裏面を使用しても構わない。その際は裏面にも解答した旨を表面に記入すること。
6. 答案用紙の指定された箇所に、科目名の「知能機械情報学（科目）」、修士・博士の別、受験番号、その答案用紙で解答する問題番号を記入すること。これらが記入漏れの場合は採点されないことがある。
7. 解答に関係のない記号や符号を記入した答案は無効となる。
8. 答案用紙は、解答ができなかった問題についても、科目名、修士・博士の別、受験番号、問題番号を記入し、2枚全部を提出すること。
9. 下書きは問題冊子の草稿用のページを用いること。
10. この問題冊子にも受験番号を記入し提出すること。

受験番号	
------	--

上欄に受験番号を記入すること。

草稿用紙  
(切り取らないこと)

草稿用紙  
(切り取らないこと)

### 問題 1 (必答問題)

問 1. 以下の問に答えよ.

- (1) 図 1 に示す回路で, 時刻  $t = 0$  でスイッチがオンになった.  $t = 0$  でコンデンサの電荷量は  $q_0$  であった. 入力電圧  $v_{IN}$  が一定のとき, 出力電圧  $v_{OUT}$  を  $t$  の関数として求めよ.

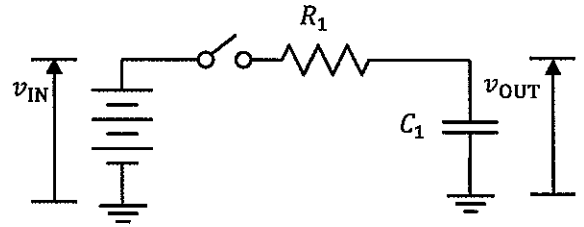


図 1

- (2) 図 2 に示す回路において, A に 1[kHz] の矩形波を与えると, B における出力波形の概形を示せ. さらに, この回路の機能の概要を説明せよ. ただし  $C_2 = 47$  [pF],  $R_2 = 10$  [k $\Omega$ ] とする.

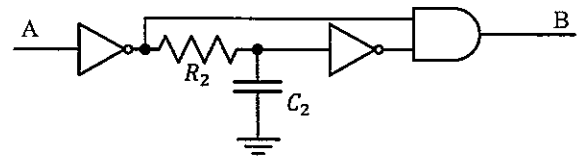


図 2

問 2. 以下の問に答えよ.

- (1) 次の伝達関数  $G(s)$  で定義されるシステムのインパルス応答とステップ応答を求めよ.  $T > 0$  は定数である.

$$G(s) = \frac{1}{1 + sT}$$

- (2) 次の状態空間モデルの可制御性と可観測性を調べよ.

$$\begin{aligned} \dot{x}(t) &= \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix} x(t) + \begin{pmatrix} 1 \\ 1 \end{pmatrix} u(t) \\ y(t) &= (1 \quad 0)x(t) \end{aligned}$$

- 問 3. 入力  $x$  から出力  $y$  を予測する線形回帰を考える. 図 3 に示す点  $(x_1, y_1), (x_2, y_2), (x_3, y_3)$  が与えられるとき,

$$\varepsilon = \sum_{i=1}^3 (y_i - wx_i - b)^2$$

を最小化する  $w$  と  $b$  を求め, そのときの  $\varepsilon$  の値を求めよ.

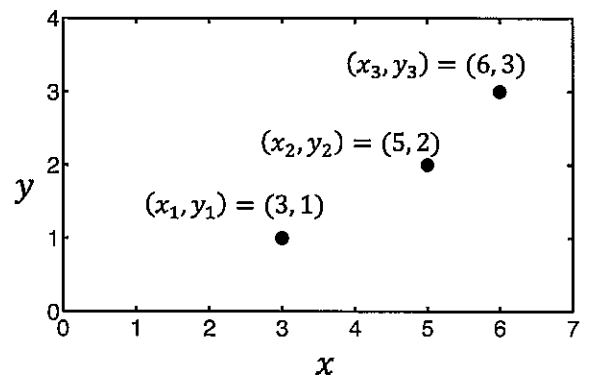


図 3

草稿用紙  
(切り取らないこと)

問題 2 A (選択問題)

図 1 に示す指示計器は水平面上に置かれており、磁界中の電流に働く力を利用した駆動トルク  $T$  により、図 2 に示す可動部を回転させる。可動部には、質量の無視できるうず巻ばねが取り付けられており、復元トルクが働く。電流が流れていないとき、指示針は図 1 の位置で静止している。指示針の静止位置からの時計回りの回転角を  $\theta$  とする。以下の問に答えよ。

問 1. 指示針を時計回りに回転させるコイル内の電流の向きを図示せよ。

問 2. コイルにステップ状の電流を印加する。可動部の回転軸まわりの慣性モーメントを  $I$ 、うず巻ばねのばね定数を  $k$ 、減衰係数を  $c$  とすると、可動部の運動は次式で表される。

$$I\ddot{\theta} + c\dot{\theta} + k\theta = T$$

- (1)  $\theta$  が振動成分をもたない臨界条件を求めよ。
- (2) 可動部は、図 2 のように、コイル、指示針、重りからなる。可動部の回転軸から重りの重心までの距離を  $d$  とする。コイルと指示針を合わせた回転軸まわりの慣性モーメントを  $I_1$ 、重りの重心まわりの慣性モーメントを  $I_2$ 、重りの質量を  $m$  とする。 $I$  を求めよ。
- (3)  $\theta$  が振動成分をもたない臨界条件を満たしている。上記の  $d$  の値を  $I_1$ 、 $I_2$ 、 $m$ 、 $k$ 、 $c$  を用いて表せ。

問 3. 一定電圧をコイルに印加し続けると、コイルの発熱による抵抗変化のため指示針が動いてしまう。そこで、図 3 のような抵抗  $R_2$ 、 $R_3$ 、 $R_4$  からなる温度補償回路を考える。図中の  $R_1$  はコイルの抵抗である。

- (1)  $R_1$  に電流  $i$  が流れたとき、電圧  $V$  と電流  $i$  の関係を  $R_1$ 、 $R_2$ 、 $R_3$ 、 $R_4$  を用いて示せ。
- (2)  $R_1$  と  $R_3$  は銅線、 $R_2$  と  $R_4$  はマンガン線とする。温度が  $\Delta t$  だけ上昇すると、銅線の抵抗値は  $(1 + a\Delta t)$  倍になるが、マンガン線の抵抗値の変化は無視できる。回路内の温度が場所によらず一様であるとき、電流  $i$  を温度変化  $\Delta t$  によらず一定に保つためには、 $R_1$ 、 $R_2$ 、 $R_3$ 、 $R_4$  がどのような関係にあればよいか。ただし、 $a\Delta t$  は微小とし、その 2 次以上の項は無視して、 $1/(1 + a\Delta t) \approx 1 - a\Delta t$  とせよ。

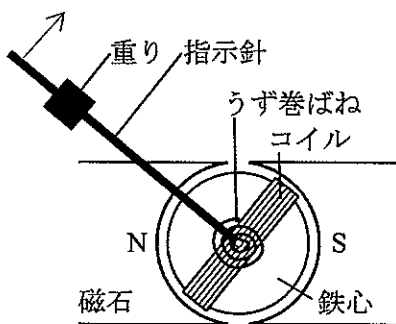


図 1

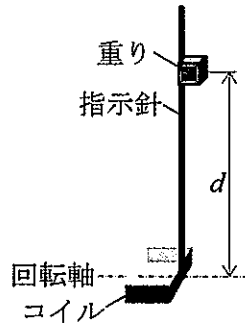


図 2

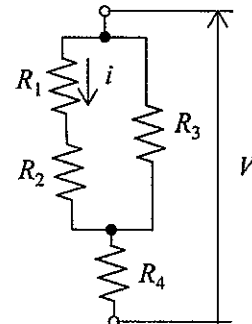


図 3

草稿用紙  
(切り取らないこと)

## 問題 2B (選択問題)

ニュートン法による平方根の計算では  $f(x)=x^2-p$  ( $p>0$ ) と置き,  $x_{k+1}=x_k-f(x_k)/f'(x_k)$  として反復的に近似解を求める. リスト1はこの方法で, ある値の平方根の計算をするC言語のプログラムであり, リスト2は等価な内容をPython言語で記述したものである. 以下の間に答えよ.

問 1. リスト1中の (A), (B) を埋め, リスト1のプログラムを実行したときの出力を3行目まで記せ.

問 2. リスト2中の (C) を埋めよ.

問 3. リスト2の関数 `my_solve` を再帰呼び出しを使って記述し直せ.

問 4. 関数 `good_enough` を以下のように記述した場合, リスト2のプログラムは正常に終了しない. 再帰呼び出しを使った `my_solve` の場合, 元の `my_solve` の場合のそれぞれで何が起こるか理由とともに記せ.

```
def good_enough(func, guess):  
    return absolute(func(guess)) == 0.0
```

問 5. リスト2の関数 `my_solve` を用いて方程式  $x^4+8x^3-8x^2-96x-110=0$  の4つの異なる実数解を全て求めるプログラムを記述せよ. ただし, 関数 `my_solve` の定義は省略してよい.

問 6. ニュートン法を用いて, 閉区間 $[0,1]$ に  $n$  個の異なる実数解を持つ  $n$  次方程式の全ての解を求めるプログラムを作成したい. そのアルゴリズムを2種類考え, 各々の概要を5行程度の文章と図を用いて記述せよ.



```

#include <stdio.h>
#include <stdlib.h>

float absolute(float x) {
    return ( x < 0 )?(-x):(x);
}

int good_enough(float (*func)(float), float guess) {
    return ( absolute((*func)(guess)) < 0.0001 );
}

float improve(float (*func)(float), float (*func_)(float), float guess) {
    if ( (*func_)(guess) == 0.0 ) exit(1);
    return _____ (A) _____ ;
}

float my_solve(float (*func)(float), float (*func_)(float), float guess) {
    while ( ! good_enough(func, guess) ) {
        printf("%.2f\n", guess);
        guess = improve(func, func_, guess);
    }
    return guess;
}

float f(float x) {
    return (x*x - 2);
}

float f_(float x) {
    return _____ (B) _____;
}

int main() {
    float a = my_solve(f, f_, 1.0);
    printf("%.2f\n", a);
    return 0;
}

```

リスト 1

```

import sys

def absolute(x):
    if x < 0:
        return -x
    else:
        return x

def good_enough(func, guess):
    return absolute(func(guess)) < 0.0001

def improve(func, func_, guess):
    func_(guess) == 0.0 and sys.exit(1)
    return guess - func(guess)/func_(guess)

def my_solve(func, func_, guess):
    while not good_enough(func, guess):
        print "%.2f"%guess
        guess = improve(func, func_, guess)
    return guess

_____ (C) _____

if __name__ == '__main__':
    a = my_solve(f, f_, 1.0)
    print "%.2f"%a

```

リスト 2

草稿用紙  
(切り取らないこと)

草稿用紙  
(切り取らないこと)

