

# 平成30年度 知能機械情報学専攻

## 大学院入学試験問題

### 「知能機械情報学（科目）」

試験日時：平成29年8月21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。

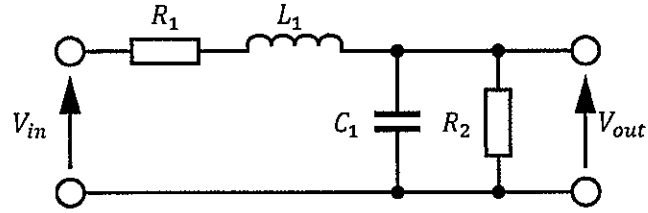
草稿用紙  
(切り取らないこと)

草稿用紙  
(切り取らないこと)

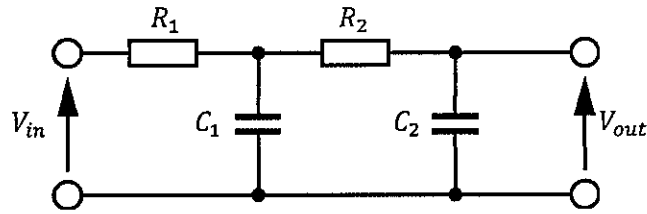
## 問題 1 (必答問題)

問 1. 電気回路について以下の問に答えよ.

(1) この回路の伝達関数を求めよ.



(2) この回路の伝達関数を求めよ.



(3) (2) で,  $R_1 = 100[\Omega]$ ,  $R_2 = 1[\text{M}\Omega]$ ,  $C_1 = 2[\text{mF}]$ ,  $C_2 = 200[\text{nF}]$  とする. 回路のボード線図を描け.

問 2. 説明変数  $x \in \mathbb{R}^d$  のサンプルを  $x_1, x_2, \dots, x_n$ , 対応する目的変数  $y \in \mathbb{R}$  のサンプルを  $y_1, y_2, \dots, y_n$  とする. 係数ベクトル  $a \in \mathbb{R}^d$  にバイアス  $b \in \mathbb{R}$  を並べたベクトル  $w = (a^\top b)^\top$  と, 説明変数  $x$  に定数 1 を並べたベクトル  $x' = (x^\top 1)^\top$  を用いて, 線形回帰モデルを  $\hat{y} = w^\top x'$  と表せるとき,  $w$  を最小二乗法で求めることができる.

(1)  $w$  を導出するための目的関数を示せ.

(2)  $w$  を導出せよ.

問 3. 以下の語句について簡潔に説明せよ.

(1) ロボティクスにおける「順運動学」と「逆運動学」.

(2) アルゴリズムにおける「最良優先探索」.

(3) 機構学における「遊星歯車」.

草稿用紙  
(切り取らないこと)

## 問題 2 A (選択問題)

図1のような1リンクのロボットアームを考える。リンクの回転関節は、減速比1:nの減速機を通してDCモータで駆動される。リンクに加わる反時計回りのトルクを $\tau$ 、鉛直軸からのリンクの角度を $\theta$ 、リンクの質量を $m$ 、リンクの回転軸まわりの慣性モーメントを $I$ 、回転軸と重心の距離を $H$ とする。重力加速度は $g$ とする。モータの出力トルクは $\tau_M$ 、入力電圧は $u$ 、逆起電力は $e_A$ 、電機子電流は $i_A$ 、モータの回転角は $\theta_M$ とする。減速機および電機子の慣性モーメントは無視できるものとする。モータのトルク定数は $K_T$ 、逆起電力定数は $K_E$ 、電機子抵抗は $R_A$ とする。電機子インダクタンスは無視できるものとする。摩擦は無視できるものとする。減速機を用いると $n\dot{\theta} = \dot{\theta}_M$ が成り立つ。

問1. ロボットアームの運動に関する以下の問に答えよ。

- (1) ラグランジュ法を用いて、 $\tau$ および $\theta$ で表されるロボットアームの運動方程式を導出せよ。
- (2)  $\tau_M$ と $\tau$ の関係を表せ。
- (3) 前問(1)で求めた運動方程式を $\tau_M$ および $\theta_M$ を使って書き直せ。
- (4) 減速機を使う場合と、減速機を使わない場合( $n=1$ )とで、ロボットアームの慣性モーメントはモータの回転にどのように影響するか、運動方程式の慣性項に注目して述べよ。

問2. ロボットアームを駆動するDCモータに関する以下の問に答えよ。

- (1)  $\tau_M$ を $K_T$ を用いて表せ。
- (2)  $e_A$ を $K_E$ を用いて表せ。
- (3) モータの等価回路は図2で表される。 $u$ を図2の他の変数で表せ。
- (4)  $\tau_M$ を $u$ および $\dot{\theta}_M$ の関数として表せ。

問3. 重力の影響を除くため、ロボットアームを水平に置く。このとき、ロボットアームの制御に関する以下の問に答えよ。

- (1) ロボットアームの状態方程式を求めよ。ただし、状態ベクトルを $x = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}$ 、入力を $u$ とする。
- (2) 前問(1)で記述したシステムの可制御性を調べよ。
- (3) 状態フィードバック $u = -K_p\theta$ を適用する。 $K_p$ は正の実数とする。 $K_p$ を大きくするとロボットアームの運動は振動的になりやすい。この振動は、減速比が十分に大きければ抑えられる。この理由をシステムの特性格方程式の解から説明せよ。

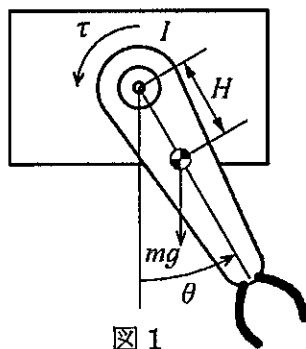


図1

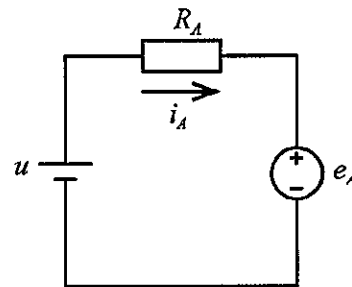


図2

草稿用紙  
(切り取らないこと)

## 問題 2B (選択問題)

問1. データ構造におけるスタックとキューに関して1, 2行程度で説明せよ。図を用いても良い。

問2. 以下の問に答えよ。リスト1-5はC言語で書かれている。

- (1) リスト1のプログラムを実行した際の出力を記せ。stack.hには関数 push(), pop(), empty\_s() が定義されている。push(), pop() はスタック操作を行い, empty\_s() はスタックが空のときに真を返す。
- (2) リスト2のプログラムを実行した際の for 文のループ毎の ep, e1, e2, e3 の関係を図を用いて説明せよ。
- (3) リスト3がスタックとして振る舞うように, プログラム中の (A), (B) を埋めよ。ただし (A), (B) は各1行である。
- (4) キューを扱う関数 enqueue(), dequeue() ならびにキューが空のときに真を返す関数 empty\_q() を定義した queue.h を記せ。ただしリスト5の element.h を利用すること。
- (5) 逆ポーランド記法とは演算子を被演算子の後に置く後置記法である。この記法では, 1と2の加算を一般の中置記法である  $1 + 2$  ではなく,  $1\ 2\ +$  と記す。中置記法で表される以下の式を逆ポーランド記法で記せ。

$$(3 - 4) * 5$$

$$(3 * 6) + (4 / 2)$$

- (6) 逆ポーランド記法を用いることで式の計算が容易に実装できる。すなわち, 順番に入力から文字を読み, それが数値であればスタックにプッシュし, 演算子 (ここでは  $+ - * /$  の二項演算子を仮定する) であれば, スタックから数値を2つポップし, その演算結果をスタックにプッシュする。これらの操作を入力が空になるまで繰り返し, スタックに残った数値が計算の結果となる。リスト4は式の数値が一桁の自然数かゼロの場合に上記を実装したプログラムである。7-13行, 14-15行, 16行, 17行の意図をそれぞれ説明せよ。また, リスト4の (C) を埋めよ。isdigit() は引数が数字であれば真を返す関数である。



```

#include <stdio.h>
#include "stack.h"

int main() {
    push('o');
    push('y');
    push('k');
    push('o');
    push('t');
    while (!empty_s()) {
        printf("%c ", pop());
    }
    return 0;
}

```

リスト 1 (main1.c)

```

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

int main() {
    ELEM e1,e2,e3, *ep;
    e1.v = '9';
    e2.v = '8';
    e3.v = '+';
    e1.p = &e2;
    e2.p = &e3;
    e3.p = NULL;
    for (ep = &e1; ep != NULL; ep = ep->p) {
        printf("%c ", ep->v);
    }
    printf("\n");
    return 0;
}

```

リスト 2 (main2.c)

```

#include <stdlib.h>
#include "element.h"
ELEM *root_s = NULL;

int empty_s () {
    return (root_s == NULL);
}

void push(char x) {
    ELEM *e;
    e = (ELEM *)malloc(sizeof(ELEM));
    if (e != NULL) {
        e->v=x;
        (A)
        root_s=e;
    }
}

char pop() {
    char r = 0;
    if(!empty_s()){
        ELEM *e;
        e=root_s;
        r=root_s->v;
        (B)
        free(e);
    }
    return r;
}

```

リスト 3 (stack.h)

```

1: #include <stdio.h>
2: #include <ctype.h>
3: #include "stack.h"
4: #include "queue.h"
5:
6: int main() {
7:     enqueue('9');
8:     enqueue('8');
9:     enqueue('+');
10:    enqueue('7');
11:    enqueue('*');
12:    enqueue('6');
13:    enqueue('-');
14:    while (!empty_q()) {
15:        char c = dequeue();
16:        if ( isdigit(c) ) {
17:            push(c-'0');
18:        }
19:    }
20:    printf("%d\n", pop());
21:    return 0;
22: }

```

リスト 4 (main3.c)

```

#ifndef _ELEMENT_H_
#define _ELEMENT_H_
typedef struct element {
    char v;
    struct element *p;
} ELEM;
#endif

```

リスト 5 (element.h)

草稿用紙  
(切り取らないこと)

## Mechano-Informatics (Subject)

Date : 2017, August 21 (Mon), 14:00 – 16:00

### Instructions:

- 0) Answers should be written either in Japanese or English.
- 1) Do not open this problem booklet until the start of the examination is announced.
- 2) Three problems are provided. Solve Problem 1 (Compulsory), and solve either Problem 2A or Problem 2B (Required Elective).
- 3) When you have multiple interpretations of a problem statement, you may clarify your interpretation by introducing adequate definitions and/or conditions in your answer.
- 4) If you find missing, misplaced, and/or unclearly printed pages in the problem booklet, notify the examiner.
- 5) Two answer sheets are provided. Check the number of them, and if you find excess or deficiency, notify the examiner. You must use a separate sheet for each problem. When you run short of space for your answer on the front side of the answer sheet, you may use the back side by clearly stating so in the front side.
- 6) In the designated blanks at the top of each answer sheet, write examination name “Mechano-Informatics (Subject)”, “Master” or “Doctor”, your applicant number, and the problem number. Failure to fill up these blanks may void your test score.
- 7) An answer sheet is regarded as invalid if you write marks and/or symbols unrelated to the answer.
- 8) Even if the answer sheet(s) is blank, submit all answer sheets with examination name, “Master” or “Doctor”, your applicant number, and the problem number.
- 9) Use the blank pages in the problem booklet for your draft.
- 10) Fill in the blank below with your applicant number, and submit this booklet.

Applicant number:
-------------------

MEMO

(Do not detach this page)

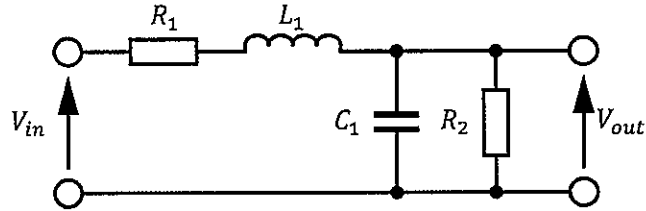
MEMO

(Do not detach this page)

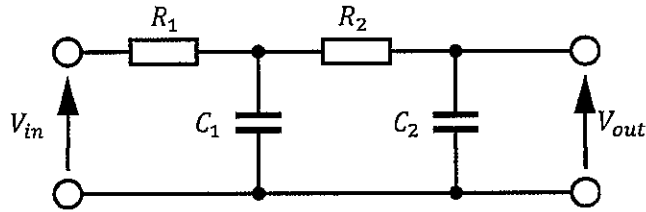
## Problem 1 (Compulsory)

P. 1. Answer the following questions about electric circuits.

(1) Derive the transfer function for this circuit.



(2) Derive the transfer function for this circuit.



(3) Let  $R_1 = 100[\Omega]$ ,  $R_2 = 1[\text{M}\Omega]$ ,  $C_1 = 2[\text{mF}]$ ,  $C_2 = 200[\text{nF}]$  in (2). Draw the Bode plot for the circuit.

P. 2. Let  $x_1, x_2, \dots, x_n$  be samples of an explanatory variable  $x \in \mathbb{R}^d$ , and  $y_1, y_2, \dots, y_n$  be samples of the response variable  $y \in \mathbb{R}$ . Using a vector  $\mathbf{w} = (\mathbf{a}^\top \ b)^\top$  consisting of a coefficient vector  $\mathbf{a} \in \mathbb{R}^d$  and bias  $b \in \mathbb{R}$ , and a vector  $\mathbf{x}' = (\mathbf{x}^\top \ 1)^\top$  consisting of the explanatory variable  $\mathbf{x}$  and a constant 1, a linear regression model can be expressed as  $\hat{y} = \mathbf{w}^\top \mathbf{x}'$ . Here  $\mathbf{w}$  can be obtained using the least squares method.

(1) Define an objective function for deriving  $\mathbf{w}$ .

(2) Derive  $\mathbf{w}$ .

P. 3. Explain the following terms briefly.

(1) "Forward Kinematics" and "Inverse Kinematics" in robotics.

(2) "Best First Search" in algorithms.

(3) "Planetary Gear" in mechanisms.

MEMO

(Do not detach this page)

## Problem 2A (Required Elective)

Consider a single-link robot arm shown in Fig. 1. The rotational joint of the link is driven by a DC motor through a gear reducer with a reduction ratio of  $1:n$ . The torque applied to the link is  $\tau$  in the counterclockwise direction, the angle of the link is  $\theta$  with respect to the vertical axis, the mass of the link is  $m$ , the moment of inertia of the link about the axis of rotation is  $I$ , and the distance between the axis of rotation and the center of mass is  $H$ . The gravitational acceleration is  $g$ . The output torque of the motor is  $\tau_M$ , the input voltage is  $u$ , the back electromotive force is  $e_A$ , the armature current is  $i_A$ , and the rotational angle of the motor is  $\theta_M$ . The moments of inertia of the gear reducer and the armature are negligible. The torque constant of the motor is  $K_T$ , the back electromotive force constant is  $K_E$ , and the armature resistance is  $R_A$ . The armature inductance is negligible. Friction is negligible. The gear reducer provides the equality  $n\dot{\theta} = \dot{\theta}_M$ .

P. 1. Answer the following questions regarding the motion of the robot arm.

- (1) Derive an equation of motion of the robot arm with  $\tau$  and  $\theta$  by using the Lagrangian.
- (2) Describe the relation between  $\tau_M$  and  $\tau$ .
- (3) Rewrite the equation of motion derived in (1) with  $\tau_M$  and  $\theta_M$ .
- (4) Describe how the moment of inertia of the robot arm affects the motor rotation when using the gear reducer and not using the gear reducer ( $n = 1$ ), by referring to the inertial term of the equation of motion.

P. 2. Answer the following questions regarding the DC motor that drives the robot arm.

- (1) Describe  $\tau_M$  with  $K_T$ .
- (2) Describe  $e_A$  with  $K_E$ .
- (3) An equivalent circuit of the motor is shown in Fig. 2. Describe  $u$  with the other variables in Fig. 2.
- (4) Describe  $\tau_M$  as a function of  $u$  and  $\dot{\theta}_M$ .

P. 3. To eliminate the effect of gravity, the robot arm is placed horizontally. Answer the following questions regarding the control of the robot arm.

- (1) Obtain an equation of state of the robot arm. Note that the state vector is  $x = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}$  and the input is  $u$ .
- (2) Examine the controllability of the system described in (1).
- (3) A state feedback  $u = -K_p\theta$  is applied. Assume that  $K_p$  is a positive real number. A large  $K_p$  tends to cause oscillatory motion of the robot arm. This oscillation can be suppressed when the reduction ratio is large enough. Explain the reason for this effect with reference to the roots of the characteristic equation of the system.

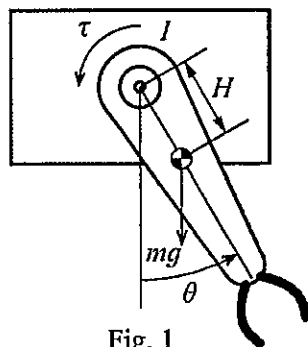


Fig. 1

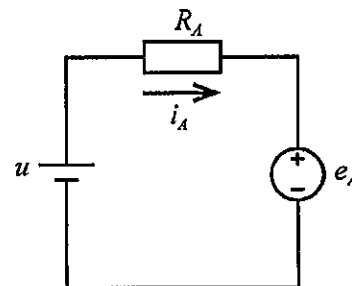


Fig. 2



MEMO

(Do not detach this page)

## Problem 2B (Required Elective)

P.1. Explain a stack and a queue in data structures in about one or two lines. You may use diagrams if needed.

P.2. Answer the following questions. Lists 1-5 are written in C.

- (1) Write the output of the program of List 1, when it is executed. Functions `push()`, `pop()` and `empty_s()` are defined in `stack.h`. Functions `push()` and `pop()` perform stack operations. Function `empty_s()` returns true when the stack is empty.
- (2) Explain the relations among `ep`, `e1`, `e2` and `e3` in each 'for' loop by using diagrams, when the program of List 2 is executed.
- (3) Fill in (A) and (B) of List 3 in one line each, so that it behaves as a stack.
- (4) Write `queue.h` that defines functions `enqueue()` and `dequeue()` which manipulate a queue, and function `empty_q()` that returns true when the queue is empty. You have to use `element.h` of List 5.
- (5) Reverse Polish Notation is a postfix notation, in which every operator follows all of its operands. The notation denotes "add 1 and 2" as `1 2 +`, instead of `1 + 2` in infix notation commonly used. Rewrite the following expressions in infix notation into Reverse Polish Notation.

`( 3 - 4 ) * 5`

`( 3 * 6 ) + ( 4 / 2 )`

- (6) Reverse Polish Notation enables us to implement a calculation of an expression easily as follows: The input tokens are read in order; if the token is a numerical value, then push it on a stack; if the token is an operator, where it is assumed to be a binary operator `+ - * /`, then pop the two values from the stack and push the calculated value on the stack. These operations are repeated until the input is empty, and the remaining value in the stack is the result of the calculation. The program of List 4 is an implementation of the above procedure for the case that the numerical values in the input are either one digit natural number or zero. Explain the meanings of the 7th-13th lines, 14th-15th lines, 16th line, and 17th line, each. Fill in (C) of List 4. Function `isdigit()` returns true if its argument is a numeric character.

```

#include <stdio.h>
#include "stack.h"

int main() {
    push('o');
    push('y');
    push('k');
    push('o');
    push('t');
    while (!empty_s()) {
        printf("%c ", pop());
    }
    return 0;
}

```

List 1 (main1.c)

```

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

int main() {
    ELEM e1,e2,e3, *ep;
    e1.v = '9';
    e2.v = '8';
    e3.v = '+';
    e1.p = &e2;
    e2.p = &e3;
    e3.p = NULL;
    for (ep = &e1; ep != NULL; ep = ep->p) {
        printf("%c ", ep->v);
    }
    printf("\n");
    return 0;
}

```

List 2 (main2.c)

```

#include <stdlib.h>
#include "element.h"
ELEM *root_s = NULL;

int empty_s () {
    return (root_s == NULL);
}

void push(char x) {
    ELEM *e;
    e = (ELEM *)malloc(sizeof(ELEM));
    if (e != NULL) {
        e->v=x;
        (A)
        root_s=e;
    }
}

char pop() {
    char r = 0;
    if(!empty_s()){
        ELEM *e;
        e=root_s;
        r=root_s->v;
        (B)
        free(e);
    }
    return r;
}

```

List 3 (stack.h)

```

1: #include <stdio.h>
2: #include <ctype.h>
3: #include "stack.h"
4: #include "queue.h"
5:
6: int main() {
7:     enqueue('9');
8:     enqueue('8');
9:     enqueue('+');
10:    enqueue('7');
11:    enqueue('*');
12:    enqueue('6');
13:    enqueue('-');
14:    while (!empty_q()) {
15:        char c = dequeue();
16:        if ( isdigit(c) ) {
17:            push(c-'0');
            (C)
        }
        printf("%d\n", pop());
    }
    return 0;
}

```

List 4 (main3.c)

```

#ifndef _ELEMENT_H_
#define _ELEMENT_H_
typedef struct element {
    char v;
    struct element *p;
} ELEM;
#endif

```

List 5 (element.h)

MEMO

(Do not detach this page)