

# 平成25年度 知能機械情報学専攻

## 大学院博士課程入学試験問題

### 「知能機械情報学（科目）」

試験日時：平成24年8月20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2 Aおよび問題2 B（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. ソフトウェアシステムにおける (1) インタプリタ, (2) コンパイラ, (3) ジャストインタイムコンパイラ, についてそれぞれの特色を説明せよ. また, 代表的なプログラミング言語を 2 つずつあげよ.

問 2. 理想的なオペアンプを用いた図 1 に示す回路について, 増幅率 ( $V_{out}/V_{in}$ ) を  $R_1, R_2, R_3$  を用いて導出せよ.

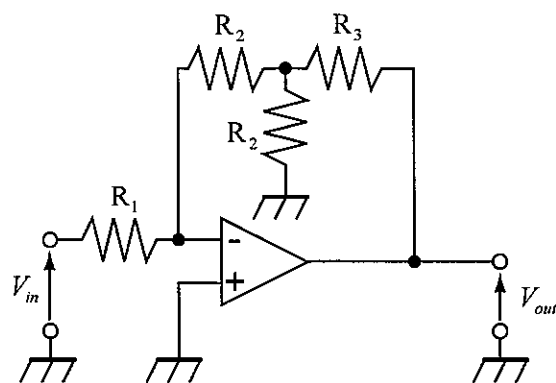


図 1

問 3. 二次元画像センサを用いて距離画像もしくは深度画像を得るための異なる 3 種類の方法をあげ, それぞれの原理と特色を述べよ.

草稿用紙  
(切り取らないこと)

## 問題 2 A (選択問題)

図 1 は、移動する目標に対して一定の距離をあけて追従する台車の実験装置である。直流モータの出力軸と台車の駆動輪との間の滑りはなく、また駆動輪と床との間の滑りもないとする。摩擦による損失は十分小さく、無視することができる。台車には距離センサが取り付けられ、目標との距離を計測する。台車の質量を  $m$ 、直流モータの出力軸の半径を  $r$ 、直流モータのコイルの抵抗を  $R$  とする。時刻  $t$  における台車の位置を  $x(t)$ 、直流モータの回転角を  $\theta(t)$ 、直流モータへの入力電圧を  $v(t)$  とする。直流モータはコイルに流れる電流に比例するトルクを発生し、その比例定数は  $K_a$  である。また直流モータの回転中には角速度に比例する逆起電力が発生し、その比例定数は  $K_e$  である。コイルを流れる電流の変化により生じる逆起電力は十分小さいとする。

- 問 1. 直流モータの入力電圧の最大値を  $V_{\max}$  とするとき、直流モータの最大の発生トルクおよび無負荷時の最大の回転角速度はそれぞれいくらか。
- 問 2. 直流モータの回転角  $\theta(t)$  と台車の位置  $x(t)$  との関係を求めよ。ただし  $\theta(0)=0$ 、 $x(0)=0$  とし、直流モータが正方向に回転すると台車も正方向に移動するとする。
- 問 3. 直流モータの負荷となる慣性モーメントはいくらか。ただし駆動輪と非駆動輪の質量、および直流モータ自身の慣性モーメントは十分小さいとする。
- 問 4. 直流モータの入力電圧  $v(t)$  と台車の位置  $x(t)$  との間の伝達関数は、 $G_c(s) = \frac{X(s)}{V(s)} = \frac{1}{s(as+b)}$  のように表すことができる。ただし  $V(s)$  と  $X(s)$  はそれぞれ  $v(t)$  と  $x(t)$  のラプラス変換である。  $a$  および  $b$  を求めよ。

目標に対して  $0.1[\text{m}]$  の間隔をあけて追従するよう台車を制御するために、図 2 のような制御系を構成した。比例ゲインを  $K_p$ 、距離センサの感度を  $K_d$  とする。問 4 の  $G_c(s)$  において  $a=0.2$ 、 $b=10$ 、また  $K_d=20$  として、以下の間に答えよ。

- 問 5. 目標がステップ状に移動したとき、オーバーシュートせずに最も速く追従する  $K_p$  の値を求めよ。
- 問 6. 問 5 の  $K_p$  のとき、目標の運動が角周波数  $0.5[\text{rad/s}]$ 、振幅  $0.05[\text{m}]$  の単振動である場合の追従誤差はおよそいくらか。単位とともに記せ。

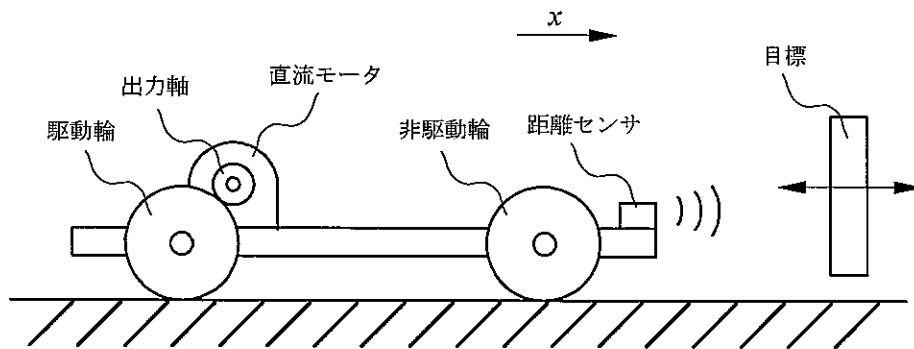


図 1

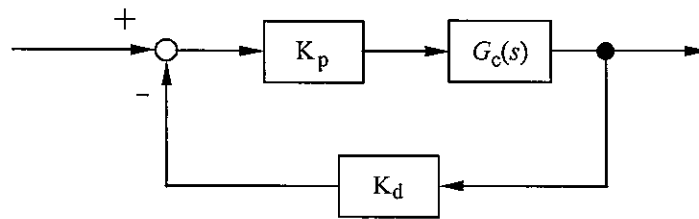


図 2

## 問題 2 B (選択問題)

リスト 1 は整数値を要素とする二分木に、以下の 2 つの制約を加えた二分ヒープと呼ばれるデータ構造を用いたプログラムである。

制約 1 : 子要素は親要素より常に大きいか等しい。

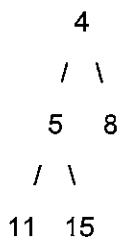
制約 2 : 深さ  $n$  の要素が全て使われるまで深さ  $n+1$  の要素は作成しない。

以下の問に答えよ。

問 1. 二分ヒープへの要素の追加のアルゴリズムは、

- 二分ヒープの最下層に要素  $e$  を追加する。
- 追加要素  $e$  とその親要素を比較する。正しい順序で並んでいるならば停止する。
- もし順序が正しくないならば、親要素と追加要素  $e$  を交換して b に戻る。

となる。以下の図で示される二分ヒープに対して、整数値 3 を要素として追加したとき、アルゴリズムが停止するまでの二分木の変化を段階を追って図で示せ。



問 2. リスト 1 のプログラムを実行した際、リスト 1 の 39 行目の出力を記せ。

問 3. リスト 1 では二分ヒープの構造がどのようにメモリ上で表現されているか。図も使って説明し、そのメリットを記せ。

問 4. 二分ヒープからルートの要素を削除するアルゴリズムを以下のようにした場合、二分ヒープの制約を保てないことがある。その理由を説明せよ。

- 二分ヒープのルートの要素を削除し、最下層の任意の要素  $e$  をルートに移動する。
- 要素  $e$  と全ての子要素を比較する。正しい順序で並んでいるならば停止する。
- もし順序が正しくないならば、要素  $e$  とその任意の子要素を交換して b に戻る。

問 5. 関数 `heap_del` は二分ヒープの制約を保ったままルートの要素を削除する関数である。関数 `heap_del_loop` 内の空白(A)~(E)を埋めよ。

問 6. リスト 1 のプログラムを実行した際の出力を記せ。また、このプログラムと同機能を実現する別のアルゴリズムの概略を記し、双方のアルゴリズムのメリット・デメリットを記述せよ。

問 7. 関数 `heap_add` を再帰を使い書き直せ。



```

1 #include <stdio.h>
2 void heap_add (int *num, int length, int c) {
3     int p, tmp;
4     if (c <= 0 || c >=length) return;
5     while (1) {
6         p = (c-1)/2;
7         if (p < 0)
8             break;
9         if (num[p] <= num[c])
10            break;
11        tmp=num[p];num[p]=num[c];num[c]=tmp;
12        c = p;
13    }
14 }
15 void heap_del_loop (int *num, int length, int p) {
16     int c, tmp;
17     while (1) {
18         c = _____ (A) _____;
19         if (c >= length)
20             break;
21         if (c+1 < length && _____ (B) _____)
22             c = _____ (C) _____;
23         if (_____ (D) _____)
24             break;
25         tmp=num[p];num[p]=num[c];num[c]=tmp;
26         p = _____ (E) _____;
27     }
28 }
29 void heap_del (int *num, int length) {
30     int p = 0, tmp;
31     tmp=num[p];num[p]=num[length];num[length]=tmp;
32     heap_del_loop(num, length, p);
33 }
34 int main () {
35     int num[10] = {2,3,0,9,4,7,8,1,6,5};
36     int length = 10, i;
37     for (i = 0; i < length; i++) printf("%d ", num[i]); printf("\n");
38     for (i = 0; i < length; i++) heap_add(num, length, i);
39     for (i = 0; i < length; i++) printf("%d ", num[i]); printf("\n");
40     while (--length >= 0) {
41         printf("%d ", num[0]);
42         heap_del(num, length);
43     }
44     printf("\n");
45     return 0;
46 }

```

リスト 1

草稿用紙  
(切り取らないこと)

草稿用紙  
(切り取らないこと)