平成１９年度 知能機械情報学専攻

大学院修士課程入学試験問題

「専門科目」

試験日時：平成１８年8月２２日（火）９：００〜１１：３０

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４題出題されており，その中から２題解答すること．
３．問題冊子の落丁，乱丁，あるいは印刷不鮮明な箇所があれば申し出ること．
４．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．
５．答案用紙の指定された箇所に，科目名の「専門科目」，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．ただし，答案用紙欄にある「枚／枚中」の記入は不要である．
６．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．
７．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枚全てを提出すること．
８．下書きは問題冊子の草稿用のページを用いること．
９．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

図1のようなライントレースロボットを考える．質量 m_1 のおもりと質量 m_2 の台が平衡状態になった位置からの変位をそれぞれ，x_1，x_2 とする．台にはフォトトランジスタ（PT_1，PT_2）と照明のための発光ダイオード（LED）が取り付けられている．おもりと台以外の質量は無視できるとする．

問1．モータの偏心などにより，走行中に車体が振動した場合のセンサの動きを考える．車体を2自由度の振動系としてとらえ，車体に対して，おもりと台は上下運動のみ可能とする場合，以下の問に答えよ．ただし，車体の振動として，外部変位 $d(t)$ が与えられると仮定する．また，k_1，$k_2/2$ はバネ定数，c は減衰係数を表すとする．

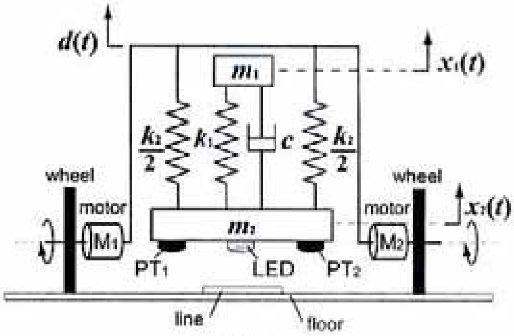

図1

（1） この振動系に対する連立運動方程式を作れ．

（2） （1）で $d(t)$ を入力，$x_1(t)$，$x_2(t)$ を出力とした場合，各々の伝達関数 G_1 と G_2 を求めよ．

（3） $d(t)=\sin\omega t$ として強制振動のみを考える．おもりと台の振幅は，それぞれ $|G_1(j\omega)|$ と $|G_2(j\omega)|$ で与えられる．$c=0$ のとき，センサの振幅が0となる ω を求めよ．

問2．車体についているセンサとその回路について考える．フォトトランジスタの信号に混入するノイズの影響を低減できるように，図2のようなコンパレータを用いる．このセンサ回路について以下の問に答えよ．なお，このコンパレータは，最大値5V，最小値0Vを出力するものとする．

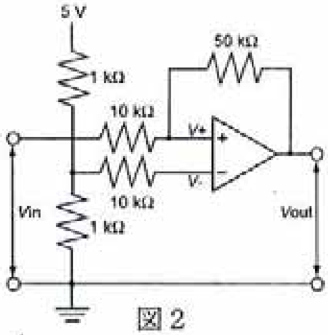

図2

（1） 図2における V_+ を Vin，Vout を用いて表せ．

（2） 今，Vout が 5V の場合，V_+ が減少し，$V_+<V_-$ となるときの Vin の範囲を求めよ．また同様に，Vout が 0V の場合，V_+ が増加し，$V_+>V_-$ となるときの Vin の範囲を求めよ．

（3） Vin に図3のような振幅 0.1V 程度の高周波ノイズの混入した信号が入力されたときの Vout について，概形を作図し，なぜそうなるのか簡単に説明せよ．

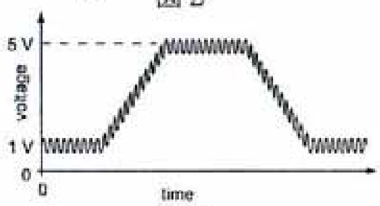

図3

問3．ライントレースロボットでは，白い床上の黒線をトレースするものと，黒い床上の白線をトレースするものが一般的である．図4はどちらの場合の動作回路か．フォトトランジスタの応答とモータの動作の関係，A, B, C の各点の電位変化を説明して答えよ．ただし，ロボットには，図2のコンパレータを持つ図4のような回路が2つ搭載され，PT_1 と M_1，PT_2 と M_2 の組み合わせによって独立に構成されているとする．また，それぞれのモータは前進するように回路に結線されているものとする．

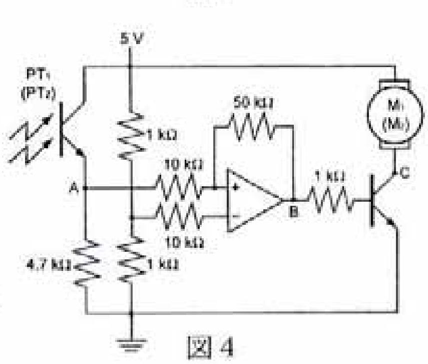

図4

以上

草稿用紙
(切り取らないこと)

問題 2

図 1 のように x 軸上の $-1.5L$, $1.5L$ の位置に 2 台の 2 関節マニピュレータの第 1 関節がそれぞれ配置されている．2 台のマニピュレータは共に第 1 リンクの長さと第 2 リンクの長さが等しく L である．図のように，左側のマニピュレータの関節角を $\theta_L = (\theta_1 \quad \theta_2)^T$ とし，エンドエフェクタの位置を $x_L = (q_{1L} \quad q_{2L})^T$，右側のマニピュレータの関節角を $\theta_R = (\theta_3 \quad \theta_4)^T$ とし，エンドエフェクタの位置を $x_R = (q_{1R} \quad q_{2R})^T$ とする．以下の各問に答えよ．

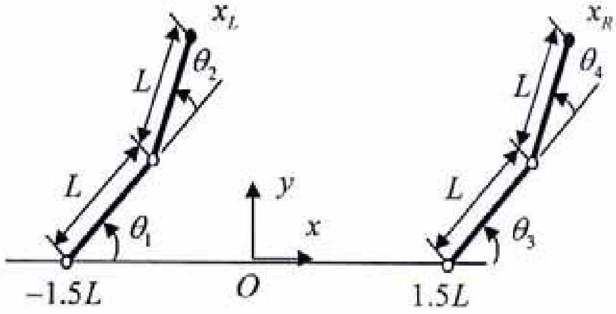

図1

問 1．$x_L = x_L(\theta_L)$ および $x_R = x_R(\theta_R)$ を θ_1，θ_2，θ_3，θ_4 で表し，ヤコビ行列 $J_L(\theta_L) = \dfrac{\partial x_L}{\partial \theta_L}$ および $J_R(\theta_R) = \dfrac{\partial x_R}{\partial \theta_R}$ を計算せよ．

問 2．図 2 のように一辺の長さが L の正方形の物体がある．正方形の中心の位置を $x_B = (q_{1B} \quad q_{2B})^T$ とし，一対の向かい合う辺の中点をそれぞれ x_{BL}，x_{BR} とする．x_{BL} から x_{BR} へ向かう直線が x 軸と成す角度を θ_B とする．$r = (q_{1B} \quad q_{2B} \quad \theta_B)^T$ とおくとき，$x_{BL} = x_{BL}(r)$ および $x_{BR} = x_{BR}(r)$ を q_{1B}，q_{2B}，θ_B で表し，ヤコビ行列 $J_{BL}(r) = \dfrac{\partial x_{BL}}{\partial r}$ および $J_{BR}(r) = \dfrac{\partial x_{BR}}{\partial r}$ を計算せよ．

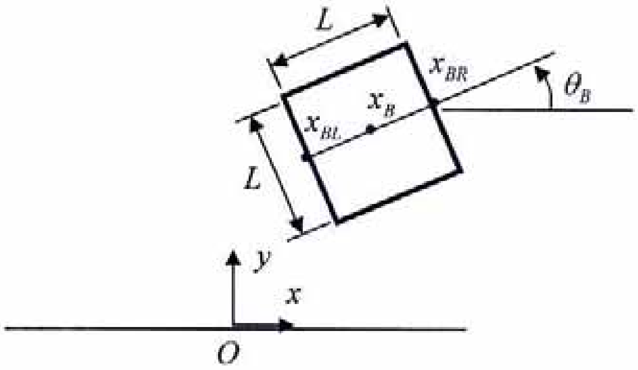

図 2

問 3．左右のマニピュレータのエンドエフェクタが物体をそれぞれ x_{BL} と x_{BR} において摩擦あり点接触で拘束し，把持する．物体の速度 $\dot{r}$ と 2 台のマニピュレータの関節角速度 $\dot{\theta} = (\dot{\theta}_L{}^T \quad \dot{\theta}_R{}^T)^T = (\dot{\theta}_1 \quad \dot{\theta}_2 \quad \dot{\theta}_3 \quad \dot{\theta}_4)^T$ の関係は $A\dot{r} = B\dot{\theta}$ のように表すことができる．A と B を J_L，J_R，J_{BL}，J_{BR} を用いて最も簡単な形で表せ．ただし，A は 4×3 行列，B は 4×4 行列とする．

問 4．図 3 のように $(\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4) = (\frac{\pi}{2} \quad -\frac{\pi}{2} \quad \frac{\pi}{2} \quad \frac{\pi}{2})$ で物体を把持するとき，$L=1$ とすると $A\dot{r} = B\dot{\theta}$ の A および B は以下のようになる．マニピュレータの関節角速度 $\dot{\theta}$ を物体の速度 $\dot{r}$ で表す式を示せ．

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -\frac{1}{2} \\ 1 & 0 & 0 \\ 0 & 1 & \frac{1}{2} \end{pmatrix} \qquad B = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & -1 \end{pmatrix}$$

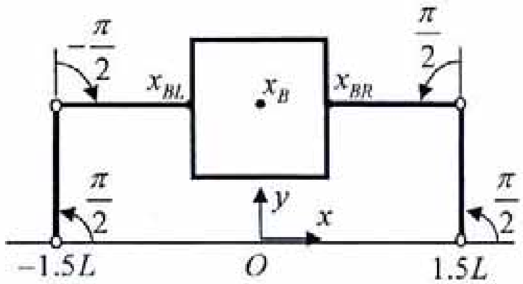

図 3

問５．問３と同様に２台のマニピュレータが物体を把持している．rank $B=4$ である．重力は存在しない．また，マニピュレータのエンドエフェクタは物体から離れたり，物体上を滑ったりしないものとする．物体の中心 x_B に外部から力 $F_B=\begin{pmatrix} f_{1B} & f_{2B} \end{pmatrix}^T$ とモーメント N_B が加わるとき，静的平衡を保つためにマニピュレータの関節が発生すべき関節トルク $\tau=(\tau_1 \quad \tau_2 \quad \tau_3 \quad \tau_4)^T$ と $F=(f_{1B} \quad f_{2B} \quad N_B)^T$ の関係を表す式を，仮想仕事の原理を用いて導け．

問６．$A\dot{r}=B\dot{\theta}$ において A のランクは物体上の把持点の位置によって定まり，B のランクはマニピュレータの姿勢によって定まる．マニピュレータのエンドエフェクタは物体から離れたり，物体上を滑ったりしないものとする．この条件を満たす $\dot{\theta}$ が $\dot{r}$ を唯一に定めるなら，マニピュレータは物体を拘束することができる．これに加えて，$\dot{\theta}$ によって任意の $\dot{r}$ を作ることができるとき，物体は「操り可能である」とよぶ．物体が操り可能である条件は，次の両式が成り立つことである．

$$\operatorname{rank} A = 3$$
$$\operatorname{Im}(A) \subseteq \operatorname{Im}(B)$$

ただし，$\operatorname{Im}(A)$ は行列 A による線形写像の像を表し，$S \subseteq T$ とは S が T の部分集合であることを表す．rank $B=4$ のとき，rank A が3から２に変化するならば，マニピュレータは物体を拘束することができるかどうか，理由とともに答えよ．また，rank $A=3$ のままで rank B が４から３に変化するとき，物体は操り可能であるかどうか，線形写像の像の関係に着目して答えよ．

以上

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3

問1．以下の項目を3行程度で説明せよ．

（1） SIMD と MIMD

（2） パイプラインハザード

（3） HDL

問2．IEEE754 では，単精度浮動小数点数は，符号部1ビット，指数部8ビット，仮数部23ビットで表される．図1のC言語プログラム test.c を実行したところ，図2のようになった．

```
#include <stdio.h>
void sub(float f)
{
   [ A ] {float f; long i;} x;
  unsigned char *p; long *l; int i; long g;
  p=(unsigned char *)&f; l = (long *)&f;
  x.f=f; g=x.i; printf(" %-12g : %x ",f,*l);
  for (i=0; i< [ B ] (float); i++) printf("%02x", *p++);
  printf(" : ");
  for (i=0; i<32; i++) {
    printf("%c", (0x80000000 & g)?'1':'0');
    if (i==0) printf(" "); else if (i==8) printf(" ");
    g=g<<1;
  }
  printf("¥n");
}
int main()
{
  float e=1,w,a=0.6, b = a+a; int i=0;
  [ C ] {
    i++; e /= 2; w = 1+e;
    printf("%2d:", i); sub(e);
  } while (w > 1);
  if ( b == 1.2 ) printf("same¥n"); else printf("diff¥n");
  sub(a); sub(0.6); sub(b); sub(1.2); sub(0.6*2); sub(2*a);
}
```

図1 test.c

```
 1: 0.5          : 3f000000 0000003f : 0 01111110 00000000000000000000000
 2: 0.25         : 3e800000 0000803e : 0 01111101 00000000000000000000000
 3: 0.125        : 3e000000 0000003e : 0 01111100 00000000000000000000000
... < 省略 > ...
22: 2.38419e-07  : 34800000 00008034 : 0 01101001 00000000000000000000000
23: 1.19209e-07  : 34000000 00000034 : 0 01101000 00000000000000000000000
24: 5.96046e-08  : 33800000 00008033 : 0 01100111 00000000000000000000000
diff
 0.6         : 3f19999a 9a99193f : 0 01111110 00110011001100110011010
 0.6         : 3f19999a 9a99193f : 0 01111110 00110011001100110011010
 1.2         : 3f99999a 9a99993f : 0 01111111 00110011001100110011010
 1.2         : 3f99999a 9a99993f : 0 01111111 00110011001100110011010
 1.2         : 3f99999a 9a99993f : 0 01111111 00110011001100110011010
 1.2         : 3f99999a 9a99993f : 0 01111111 00110011001100110011010
```

図２ test.c の実行結果

（１） 空欄 $\boxed{A}$, $\boxed{B}$, $\boxed{C}$ に入る単語を示せ.

（２） 計算機のバイトオーダとはどういうものか説明し，図２を出力した計算機におけるバイトオーダとその呼び名を示せ.

（３） test.c における while ループが終了する理由を説明せよ.

（４） 図２において，same ではなく，diff が出力される理由を説明せよ.

（５） 図２より，仮数部の最上位ビットが省略され指数部もオフセットが付けられていることがわかる．仮数部が 0 でない例を以下に示す．1.75, 4.875, 5.75 がどのようなビット列で表わされるかを示せ.

```
-1.5        : 1 01111111 10000000000000000000000
0.625       : 0 01111110 01000000000000000000000
0.75        : 0 01111110 10000000000000000000000
1.625       : 0 01111111 10100000000000000000000
4.125       : 0 10000001 00001000000000000000000
```

（６） 浮動小数点数の加算計算のアルゴリズムを説明せよ.

（７） 浮動小数点数の演算における数値計算誤差の種類とそれぞれの発生理由を説明せよ.

以上

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4

問1．次の語句をそれぞれ対比させながら3行程度で説明せよ．

（1） サーバとクライアント

（2） プロセスとスレッド

（3） 逐次実行とイベント駆動

問2．図1のように道路で結ばれた都市の集合 N={0,1,2,…,7}があるとき，都市 s から都市 g(ただし s,g∈N) まで移動するための経路を求める問題を考える．図中の道路に付記された数字は各都市間の所要時間を表し，往路・復路の所要時間はそれぞれ同じとする．

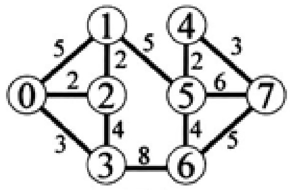

図1

s から g まで移動する経路が存在するかどうかを判定するアルゴリズムをアルゴリズム1に示す．ここで，記号←は代入を表す．また，ϕは空集合，E(k)は都市 k と1本の道路で結ばれている都市の集合，v は整数値を要素とする配列，GetFirst(Q)は集合 Q の最初の要素を取り出して Q から除く関数，Append(Q,i)は集合 Q の最後に要素 i を追加する関数である．

（1）Q のデータ構造としてリスト1の1～4行目に定義された構造体を要素とする線形リストを用いるとき，Append()関数のC言語による実装の一つをリスト1に示す．リスト1の空欄 $\boxed{\text{A}}$ ～ $\boxed{\text{C}}$ を埋めよ．なお，first はリストの先頭を指す大域変数である．

（2） s←0, g←7 としてアルゴリズム1を実行したとき，6行目を実行する直前における Q の要素を下の例にならって示せ．ただし，集合 E(k)の要素は都市番号の小さい順に並んでいるものとする．

（例）　{0}→{1, 2}→{2, 3, 4}

s から g まで移動できる経路が複数存在するとき，その最短所要時間も計算できるように改良したアルゴリズムをアルゴリズム2に示す．ここで t は整数値を要素とする配列，T(k,i)は i∈E(k)である都市 k から i までの所要時間，Insert(Q,i)は集合 Q の要素が t[i]の昇順になるように i を挿入する関数，Remove(Q,i)は要素 i を集合 Q から除く関数である．

（3） Insert()関数の実装の一つをリスト2に示す．リスト2の空欄 $\boxed{\text{D}}$ ・ $\boxed{\text{E}}$ を埋めよ．ただし，空欄 $\boxed{\text{A}}$ ～ $\boxed{\text{C}}$ には（1）と同じものが入り，構造体・変数の定義はリスト1と共通とする．

（4）（3）で完成させた Insert()関数を使い，s←0, g←7 としてアルゴリズム2を実行したとき，7行目を実行する直前における t の要素を下の例にならって示せ．

（例）　{∞, ∞, ∞, ∞, ∞, ∞, ∞, ∞}→{5, ∞, ∞, 4, ∞, ∞, 3, ∞}

（5） Insert()関数の実行速度を改善するために Q のデータ構造を改良する方法を考え，その概要を3行程度で述べよ．

（6）アルゴリズム2の結果を使って最短所要時間を与える経路を求めて配列 p[]に格納し，s, g を含む通過する都市数を返す関数 FIND_PATH(p)をアルゴリズム1，2にならって記述せよ．なお，最短所要時間を与える経路が複数ある場合は，そのうち一つを見つければよいものとする．また，必要に応じてC言語と同様の制御文を使ってよい．

```
HAVE_PATH(s,g)
    for all i∈N do v[i]←0; end.
    Q←{s};
    v[s]←1;
    while Q≠ϕ do
        k←GetFirst(Q);
        if (k=g) then
            return SUCCESS;
        end.
        for all i∈E(k) do
            if (v[i]=0) then
                v[i]←1;
                Append(Q,i);
            end.
        end.
    end.
    return FAILURE.
```

アルゴリズム1

```
MIN_TRAVEL_TIME(s,g)
    for all i∈N do t[i]←∞; v[i]←0; end.
    Q←{s};
    t[s]←0;
    v[s]←1;
    while Q≠ϕ do
        k←GetFirst(Q);
        if (k=g) then
            return t[g];
        end.
        for all i∈E(k) do
            if (v[i]=0) then
                v[i]←1;
                t[i]←t[k]+T(k,i);
                Insert(Q,i);
            end.
            else if (t[k]+T(k,i)<t[i]) then
                t[i]←t[k]+T(k,i);
                if i∈Q then
                    Remove(Q,i);
                    Insert(Q,i);
                end.
            end.
        end.
    end.
    return -1.
```

アルゴリズム2

```
typedef struct _Queue {
  int city;
  struct _Queue* next;
} Queue;

Queue* first=NULL;

void Append(int i){
  Queue *q, *c;
  q=malloc(sizeof(Queue));
  q->city=i;
  q->next=NULL;
  if(!first) {
    [ A ]=[ B ];
    return;
  }
  for(c=first;c;c=c->next){
    if(!c->next){
      c->next=[ C ];
      return;
    }
  }
}
```

リスト1

```
int t[8];

void Insert(int i){
  Queue *q, *c;
  q=malloc(sizeof(Queue));
  q->city=i;
  q->next=NULL;
  if(!first){
    [ A ]=[ B ];
    return;
  }
  if(t[q->city]<t[first->city]){
    q->next = first;
    first = q;
    return;
  }
  for(c=first; c; c=c->next){
    if(!c->next){
      c->next=[ C ];
      return;
    }
    else if(t[q->city]<t[c->next->city]){
      q->next=[ D ];
      c->next=[ E ];
      return;
    }
  }
}
```

リスト2

以上

草稿用紙

(切り取らないこと)

草稿用紙
(切り取らないこと)