

Mechano-Informatics (Subject)

Date : H24 (2012), August 20th, 14:00 – 16:00

Instruction:

- 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) Submit both answer sheets even if they are blank.
- 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. Also submit the Japanese booklet with your applicant number in the corresponding blank.

Applicant number:

MEMO
(Do not detach this page)

MEMO
(Do not detach this page)

Problem 1 (Compulsory)

- P. 1. Explain characteristics of each of the following items in a software system: (i) an interpreter, (ii) a compiler and (iii) a just-in-time compiler. Also, list two typical target programming languages of each item.
- P. 2. Assuming an ideal operational amplifier, derive the gain (V_{out}/V_{in}) of the circuit shown in Fig.1 using R_1 , R_2 , and R_3 .

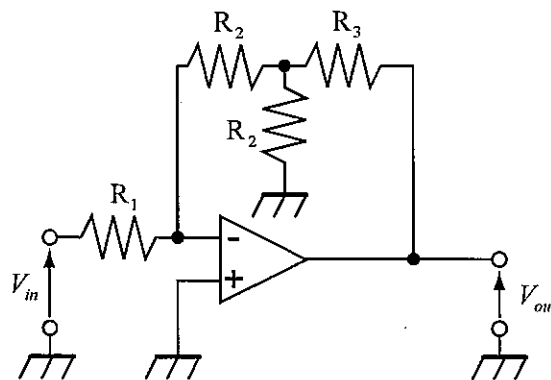


Fig.1

- P. 3. List three different methods to obtain a distance image or a depth map by using a two-dimensional image sensor(s), and explain the principles and characteristics of each method.

MEMO
(Do not detach this page)

Problem 2A (Required Elective)

Figure 1 shows an experimental setup of a carriage which is controlled to keep a certain distance to a target. No slip occurs between the output shaft of the DC motor and the driving wheels of the carriage, nor between the driving wheels and the floor. Loss due to friction is negligibly small. A distance sensor is installed on the carriage to measure the distance to the target. Let the mass of the carriage m , radius of the output shaft of the DC motor r , and resistance of the DC motor coil R . Let the position of the carriage $x(t)$ at time t , rotation angle of the DC motor $\theta(t)$, and input voltage to the DC motor $v(t)$. The DC motor generates torque proportional to the coil current, where the proportional constant is K_a . Counter electromotive force proportional to the angular velocity is generated during the DC motor rotation, where the proportional constant is K_e . Assume that the counter electromotive force induced by a change of the DC motor coil current is negligibly small.

- P. 1. Assume that the maximum input voltage to the DC motor is V_{\max} . Show the maximum torque of the DC motor, and the maximum rotational angular velocity of the DC motor in no-load running.
- P. 2. Show the relationship between the rotation angle of the DC motor $\theta(t)$ and the position of the carriage $x(t)$. Assume that $x(0)=0$ and $\theta(0)=0$, and the carriage moves in the positive direction when the DC motor rotates in the positive direction.
- P. 3. Show the inertial moment of the DC motor load. Assume that the masses of the driving wheels and non-driving wheels, and the inertial moment of the DC motor itself are negligibly small.
- P. 4. The transfer function between the input voltage of the DC motor $v(t)$ and the position of the carriage $x(t)$ can be expressed by $G_c(s) = \frac{X(s)}{V(s)} = \frac{1}{s(as+b)}$ where $V(s)$ and $X(s)$ are the Laplace transformations of $v(t)$ and $x(t)$, respectively. Derive a and b .

A control system shown in Fig. 2 is constructed to make the carriage track the target with 0.1 [m] distance. Let the proportional gain K_p and the sensitivity of the distance sensor K_d . Assume that $a = 0.2$ and $b = 10$ in $G_c(s)$ defined in P. 4, and $K_d = 20$. Solve the following problems.

- P. 5. Show the value of K_p for the fastest response without overshoot when the target moves stepwise.
- P. 6. In the case of K_p obtained in P. 5, what is the approximate value of the tracking error to the target that undergoes simple harmonic oscillation with an angular frequency of 0.5 [rad/s] and an amplitude of 0.05 [m] ? The answer must be described along with its unit.

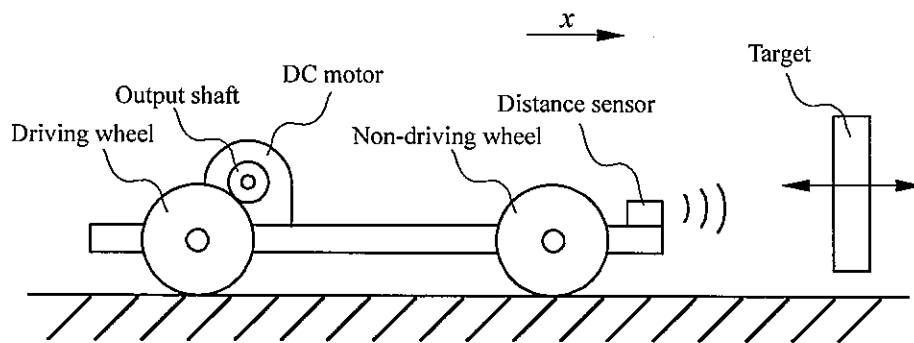


Fig. 1

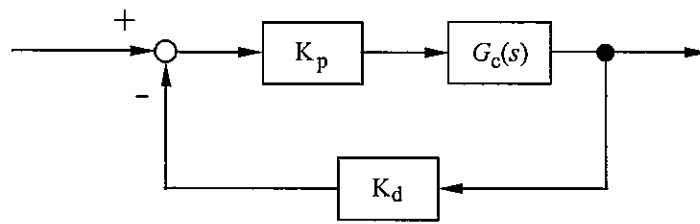


Fig. 2

Problem 2B (Required Elective)

List 1 is a program using a data structure called a binary heap, which is a binary tree of integer elements satisfying the following two constraints:

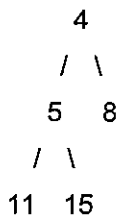
Constraint 1: A child element must be greater than or equal to its parent.

Constraint 2: An element at the $n+1$ th level of the tree is not created until the n th level is full.

Solve the following problems.

- P. 1. An algorithm to add an element e to a binary heap is written as follows:
- Add the element e to the bottom level of the binary heap.
 - Compare the added element e with its parent element. If they are in the correct order, stop.
 - If not, swap the added element e with its parent element, go to step b.

When integer number 3 is added as an element to the binary heap shown in the following figure, draw the sequence of the changes in the binary tree until the algorithm stops.



- P. 2. Describe the output of the 39th line of List 1, when the program of List 1 is executed.
- P. 3. In List 1, how is the binary heap structure represented in the memory? Explain with figures and describe the advantages.
- P. 4. When considering an algorithm that deletes the root element from the binary heap as follows, the binary heap constraints may not be maintained. Describe the reason.
- Delete the root element of the binary heap and move any element e at the bottom level to the root.
 - Compare the element e with its children. If they are in the correct order, stop.
 - If not, swap the element e with any of its child element(s), go to step b.
- P. 5. Function `heap_del` deletes the root element while maintaining the binary heap constraints. Fill in the blanks (A)-(E) in function `heap_del_loop`.
- P. 6. Describe the output of the program of List 1. Show an outline of another algorithm that performs an equivalent function. Explain the advantage(s) and disadvantage(s) of each algorithm.
- P. 7. Rewrite function `heap_add` as a recursive function.


```

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 }

```

List 1

MEMO
(Do not detach this page)

MEMO
(Do not detach this page)