Mechano-Informatics (Subject)

Date : H26(2014), August 18th, 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) 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. 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. Answer the following questions.

(1) The switch in the circuit of Fig.1 turns on at time $t = 0$. The charge on the capacitor is q_0 at $t = 0$. When the input voltage v_{IN} is constant, obtain the output voltage v_{OUT} as a function of t.

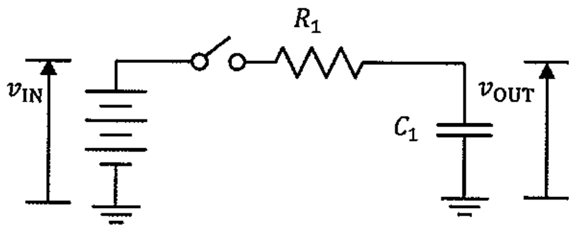

Fig.1

(2) Assume that 1[kHz] square-wave is given at A in the circuit of Fig.2. Sketch the output waveform at B in the circuit. Explain the function of this circuit briefly. We assume $C_2 = 47$ [pF], and $R_2 = 10$ [kΩ].

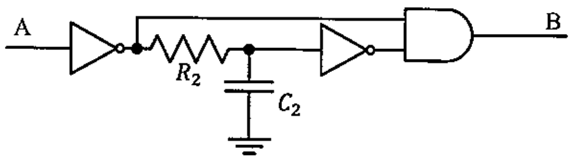

Fig.2

P. 2. Answer the following questions.

(1) Obtain the impulse response and the step response of the system defined by the following transfer function $G(s)$. Note that $T > 0$ is the constant number.

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

(2) Examine the controllability and the observability of the following state space model.

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

P. 3. Consider linear regression predicting output y from input x. When the points (x_1, y_1), (x_2, y_2), and (x_3, y_3) in Fig.3 are given, obtain the optimal w and b that minimize

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

Calculate the corresponding ε.

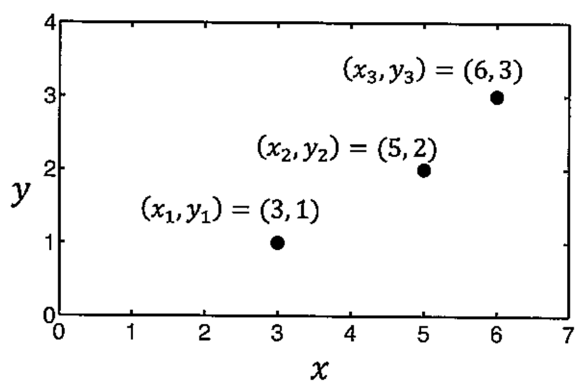

Fig.3

MEMO

(Do not detach this page)

Problem 2A (Required Elective)

An indicating instrument in Fig. 1 is placed on a horizontal plane. Driving torque T generated by a force acting on a current in a magnetic field rotates the moving part of Fig. 2. A spiral spring with negligible mass is attached to the moving part and generates restoring torque. When no current flows, the pointing needle stays at the resting position as shown in Fig. 1. The clockwise rotation angle from the resting position is denoted as θ. Answer the following questions.

P. 1. Draw the direction of current flow in the coil that rotates the pointing needle clockwise.

P. 2. Consider a step-input current applied to the coil. The moment of inertia of the moving part around the rotation axis is I, the spring constant of spiral spring is k, and the damping coefficient is c. The motion of moving part is given as follows:

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

(1) Obtain the critical condition for θ to have no oscillatory component.

(2) As shown in Fig. 2, the moving part consists of a coil, pointing needle and weight. The distance from the rotation axis of the moving part to the center of mass of the weight is d. The moment of inertia of combined structure of the coil and pointing needle around the rotation axis is I_1, the moment of inertia of weight around the center of mass is I_2, and the mass of weight is m. Obtain I.

(3) θ has met the critical condition to have no oscillatory component. Determine d in the above with I_1, I_2, m, k and c.

P. 3. When a constant voltage is continuously applied to the coil, the pointing needle drifts due to change of resistance caused by heat generation at the coil. Hence, consider a temperature compensation circuit with resistances R_2, R_3 and R_4 in Fig. 3. R_1 in the figure is the resistance of the coil.

(1) When current i flows through R_1, relate voltage V and current i in Fig. 3 using R_1, R_2, R_3 and R_4.

(2) Assume that R_1 and R_3 are made of copper and that R_2 and R_4 are made of Manganin. As the temperature increases by Δt, the resistance of copper becomes $(1+\alpha\Delta t)$ times larger, while the change of resistance in Manganin is negligible. When the temperature in the circuit is spatially uniform, describe the relationship among R_1, R_2, R_3 and R_4 in order to keep the current i constant regardless of Δt. Assuming that $\alpha\Delta t$ is small, ignore its second-order or higher terms, and use $1/(1+\alpha\Delta t)\approx 1-\alpha\Delta t$.

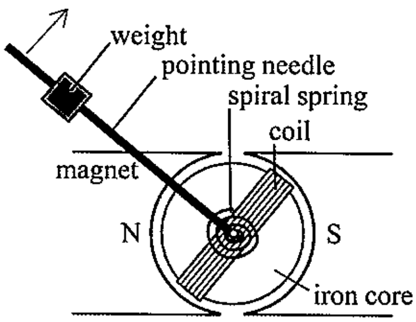

Fig. 1

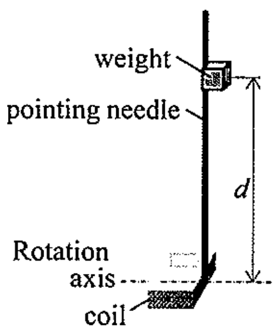

Fig. 2

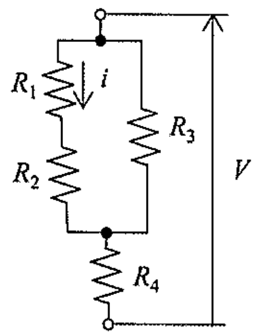

Fig. 3

MEMO

(Do not detach this page)

Problem 2B (Required Elective)

Newton's method for square root finds an approximate solution by iteratively calculating $x_{k+1} = x_k - f(x_k)/f'(x_k)$, where $f(x) = x^2 - p \quad (p > 0)$. List 1 is a program to compute square root of a certain value by using this method written in C and List 2 is an equivalent program written in Python. Answer the following questions.

P. 1. Fill in (A) and (B) of List 1. Write the first three lines of the output when this program in List 1 is executed.

P. 2. Fill in (C) of List 2.

P. 3. Redefine function `my_solve` in List 2 using a recursive call.

P. 4. The program of List 2 will not terminate successfully when function `good_enough` is defined as follows. Explain what happens in the case using `my_solve` with a recursive call and the case using the original `my_solve`. Describe the reason for each.

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

P. 5. Describe a program to find four distinct real solutions of $x^4 + 8x^3 - 8x^2 - 96x - 110 = 0$ using the function `my_solve` defined in List 2. You can omit the definition of the function `my_solve`.

P. 6. Consider writing a program to find all the solutions of a given n-th order equation that has n distinct real solutions on the closed interval [0, 1], using Newton's method. Devise two different algorithms and describe an outline of each in about 5 lines of text and figure(s).

```
#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;
}
```

List 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 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
```

List 2

MEMO

(Do not detach this page)

MEMO

(Do not detach this page)