CT.2306 : Signals & Systems II

Project - (2023-2024)

Notes :

- This project, composed of **18 questions**, is to be treated by **trinomial**, except for a few exceptions due to the number of students not being a multiple of 3.
- No changes to the list of trinomials are allowed.
- The drop boxes on Moodle will soon be created in the names of each trio. You will be informed to verify that there are no configuration problems.
- The deadline for submitting the report which will cover the answers to all the questions in this subject, as well as the Matlab/Simulink programs is **Sunday January 7, 2024 at 11 :59 p.m.**
- All Matlab/Simulink graphics are to be presented in the report.
- The Matlab/Simulink code(s)/schema(s) for each question must be presented when answering the concerned question. No appendix for all codes.
- For each Matlab figure presented, grid the figure (`grid`), label the abscissa axis (`xlabel`) and that of the ordinates (`ylabel`), and legend the curves if the figure contains two or more curves (`legend`).
- The format of the report is **.pdf** and of the programs is **.zip** or **.rar**.
- The project grade has the same coefficient as a supervised exam (so 3).
- Students who have not provided their names for the project group, and whose groups I have constructed, will have $\mathbf{-2}$ points of the project grade.
- Sending the report or programs by email, before the deadline, involves a penalty of 0.5 points of the project grade.
- Sending the report or programs by email, after the deadline, involves a penalty of 1 point + 0.5 point for each 30 minutes of delay.
- Failure to comply with the required report format (.pdf) results in a penalty of 1 point.
- Failure to comply with the format required for programs (.zip or .rar only) involves a penalty of 1 point.
- The report may be handwritten provided it is well done. A penalty of 1 point will be applied if the quality of the report is too poor.
- A report written in LaTeX will benefit from a bonus of 1 point.
- There will be no defenses of the project.

Processing motion signals from a PTZ camera

PTZ type cameras (*Pan–Tilt–Zoom*) are video cameras, capable of horizontal panning (left–right), vertical tilt (up–down) and a zoom (for magnification).

FIGURE 1 – Examples of PTZ cameras.

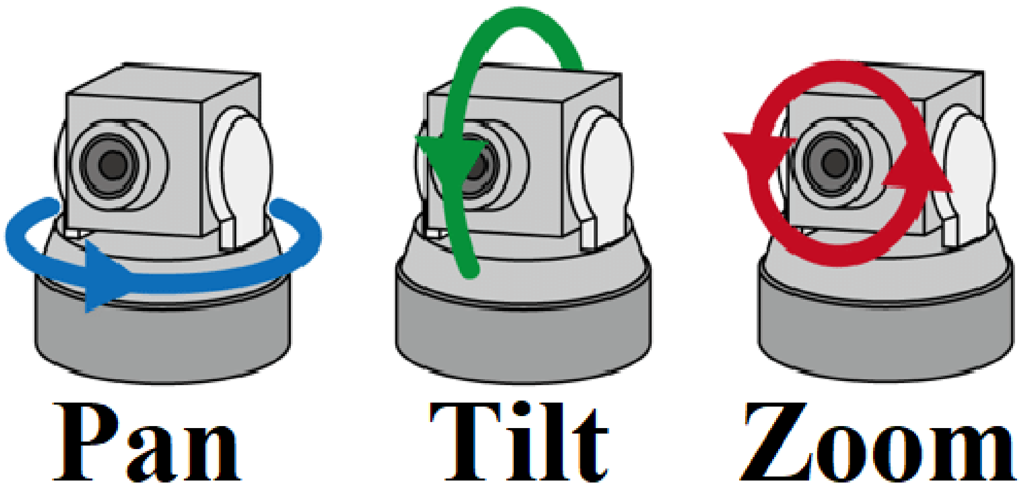

FIGURE 2 – Pan–Tilt–Zoom movements.

PTZ cameras are often positioned at guard stations where active employees can manage them using a remote camera controller. Their primary function is to monitor large open areas requiring views in a 180 or 360 degree range. These cameras are equipped with two motors providing pan and tilt movements. In order to be able to control the pan and tilt angular positions, two servos must be studied. This requires the acquisition of data on the behavior of each motor in order to be able to develop a model of the camera allowing the calculation of a command which will ensure the control of the pan and tilt movements of the camera.

During this study, we are interested in the processing of data on the angular speed of the motor ensuring the pan movement, denoted $\omega(t)$. This data was collected using an angular speed sensor installed at the pan motor. However, the study will also require data on the angular position and acceleration, denoted respectively $\theta(t)$ and $\dot{\omega}(t)$, in order to be able to elaborate, with the tilt motion data, a model known as the *state model* (or state space model), which is used to study MIMO (Multi Input-Multi Output) systems.

Data visualization

The file `data-proj.mat` contains the data of the angular speed of the motor ensuring the pan movement, measured by the sensor.

1) Download the file then upload it to Matlab Drive. Load data into the Matlab Workspace, then list the names of the vectors available in the file as well as their sizes.

2) On Matlab, plot the angular speed $\omega(t)$ as a function of time. Add a grid to the figure and label the x-axis as `Time [sec]` and the y-axis as `Angular speed [rad/sec]`.
Is it possible to use this signal as it is, directly to control the angular position of the camera ? Justify.

Analog filtering

The available data is already digitized with a sampling period denoted T_{e_1}.

In order to be able to carry out an analog−digital study, we will assume that the current data is analog that we want to digitize with a sampling period $T_{e_2} = 0.05$ *sec*.

3) Determine the sampling period T_{e_1} (it is in seconds).

4) Use one of the codes provided during workshop N°5 (DFT or FFT) to calculate then plot the amplitude spectrum (bilateral and centered) of $\omega(t)$ measured.
Use the `stem` command to plot the spectrum.
Limit frequency axis when displaying between -2 and $+2$ Hz.

5) Knowing that the frequencies contained in the $\omega(t)$ signal are less than or equal to 2 Hz and have amplitudes greater than 0.2 rad/sec, specify the frequencies contained in the signal, and deduce F_M, its maximum frequency.

6) Design a first order low-pass analog filter of a unit gain, making it possible to filter the measured signal $\omega(t)$. The signal $\omega(t)$ after filtering will be denoted $\omega_f(t)$. For that :

- Propose a cutoff frequency f_c for the filter to be designed. Calculate its cutoff pulse ω_c.
- Establish the transfer function of the filter $H_1(p)$.
- Design the filter and filter the data **on Matlab then on Simulink** :
  - **On Matlab :**
    - Use the command `tf` to create the transfer function of the filter $H_1(p)$.
    - Use the command `lsim` to filter and retrieve the filtered data $\omega_f(t)$.
    - In a new figure, overlay the graphs of $\omega(t)$ and $\omega_f(t)$ with two different colors (don't forget the legend).
  - **On Simulink :**
    - Use the `Transfer Fcn` block to design the filter transfer function.
    - The filter input is the noisy data $\omega(t)$. To inject them at the filter input in Simulink, create in Matlab a two-column matrix named for example `data` which contains the vector t as the first column and the vector ω as the second column.

Attention ! The data is provided as row vectors. To transpose them to columns we use $'$: a' is the transposed vector of a.

- Import the restructured data in the *data* matrix from the Workspace to Simulink using the `From Workspace` block. In the `From Workspace` block settings, change the data name from `simin` to `data`.
- Set the simulation step to 0.001 *sec* and the simulation duration to the duration of the measured data $\omega(t)$.
- Visualize the data after filtering on a `Scope` block.
- Overlap the data before and after filtering on the same Scope using a `Mux` block.
- Take a screenshot of the Scope screen and present it in the report.

7) Use one of the codes provided during workshop N°5 (DFT or FFT) to calculate then plot the amplitude spectrum (bilateral centered) of $\omega_f(t)$ filtered (in red), superimposed with that of $\omega(t)$ (in blue).
Use the `stem` command to plot the spectrum.
Limit the frequency axis when displaying between -100 and $+100$ Hz for a first display, then between -2 and $+2$ Hz for a second display.
Compare the spectra of $\omega(t)$ before and after filtering. Comment on the results.

Sampling

The objective of this part is to sample the filtered angular velocity vector $\omega_f(t)$ with a sampling period $T_{e_2} = 0.05$ *sec*.

8) Propose a Matlab code allowing to create a vector $\omega_e(t)$ which contains the values of the vector $\omega_f(t)$ with a period between the values of $T_{e_2} = 0.05$ *sec*.

9) What is the size of the new vector $\omega_e(t)$?

10) Create a new vector t_e which corresponds to the vector $\omega_e(t)$ (with the same number of components, and whose instants are spaced $T_{e_2} = 0.05$ *sec*).

11) Overlay on the same graph, the signal of the filtered angular velocity $\omega_f(t)$ (`plot`) and that of the sampled filtered angular velocity $\omega_e(t)$ (`stem`), with two different colors. Then, zoom between moments 10 and 12 *sec*.

12) Calculate the discrete Fourier transform of $\omega_e(t)$, then overlay on the same graph the amplitude spectrum (bilateral centered) of $\omega_e(t)$ with those of $\omega(t)$ and $\omega_f(t)$ (in different colors)
Compare the three spectra then comment on them.

Angular position and acceleration

As mentioned above, controlling the camera will require in addition to angular velocity data, corresponding angular position and acceleration data. The use of three sensors to measure these three physical quantities $\theta(t)$, $\omega(t)$ and $\dot{\omega}(t)$, is not a recommended solution for reasons the cost and volume of the device. As these three quantities have a derivative relation between them, $\omega(t) = \dfrac{d\theta(t)}{dt}$, $\dot{\omega}(t) = \ fracd\omega(t)dt$, it is recommended to install only one sensor to measure one of these quantities, then calculate the other two by derivation/integration.

13) From the filtered and sampled angular velocity data $\omega_e(t)$, calculate by derivation, the angular acceleration $\dot{\omega}(t)$ and by integration, the position $\theta(t)$. To do this, use these relations of numerical derivation and integration :

- **Angular acceleration :**
  - First component : $\dot{\omega}(1) = \dfrac{\omega_e(2) - \omega_e(1)}{Te}$.
  - Last component : $\dot{\omega}(n) = \dfrac{\omega_e(n) - \omega_e(n-1)}{Te}$.
  - The other components : $\dot{\omega}(k) = \dfrac{\omega_e(k+1) - \omega_e(k-1)}{2Te}$.
- **Angular position :** $\theta(k) = T_e \cdot \sum_{i=0}^{k} \omega_e(i)$.

The vectors $\theta(t)$ and $\dot{\omega}(t)$ have the same dimensions as the vector $\omega(t)$.

There are other numerical derivation/integration formulas, notably those seen during the Numerical Methods course.

14) Plot the graphs of $\theta(t)$ and $\dot{\omega}(t)$ (on two different figures). Comment on both graphs.

15) Calculate the discrete Fourier transforms of $\theta(t)$ and $\dot{\omega}(t)$, then plot on two different figures, the amplitude spectra (bilateral centered) of $\theta(t)$ and $\dot{\omega}(t)$.

What can we say about the degree of noise associated with $\theta(t)$ and $\dot{\omega}(t)$?

Digital filtering

The angular acceleration signal $\dot{\omega}(t)$ obtained by digital derivation is noisy compared to the other two signals. We use a digital filter to filter it. The signal $\dot{\omega}(t)$ after filtering is denoted $\dot{\omega}_f(t)$.

16) Use the bilinear transformation method to calculate the transfer function $H_2(z)$ of an IIR type digital filter from the one designed in question 6 (same cutoff frequency f_c).

17) On Matlab, filter the signal $\dot{\omega}(t)$ using the filter $H_2(z)$, and present its graph overlayed with the one before filtering.

For that :

- Declare the discrete transfer function $H_2(z)$:

  `Te = .....; z = tf('z',Te); H2 = tf([num coef],[denum coef],Te)`

- Use the `lsim` command to calculate $\dot{\omega}_f(t)$ at the filter output.

18) Calculate the discrete Fourier transform of $\dot{\omega}_f(t)$, then plot its amplitude spectrum (bilateral centered) overlayed with that of $\dot{\omega}(t)$.