

Homework #8

LabVIEW

Dr. Pogo

Assignment is due on Thursday, November 16, 2023

Assigned October 31, 2023

Tips:

1. Besides your LabVIEW code, **you should submit a video** that demonstrates you using your own program. Be sure that you demonstrate EVERY required feature above: if you don't show it in your video, then you didn't do it! Your video should be recorded from a perspective that lets me see your actual pendulum, and your onscreen controls *simultaneously*. For example, when I watch your "simulation", I want to see it side-by-side in your submitted video with the actual (partly decayed) pendulum so I can directly compare the frequencies.
2. You may use my "start video" subvi.
3. You'll need an *array* of IMAQ images to store the video data.
4. For image processing, I used the following method:
 - a. I extracted the red color plane, using "IMAQ ExtractColorPlanes". This also required creating a new IMAQ image to store the extracted plane separately from the original video. The extracted image is actually gray scale, but is "whiter" where the original image was more red.
 - b. In a "For" loop, I used "IMAQ Threshold" multiple times for each extracted image. "Thresholding" means converting a grayscale image into a black and white image. I kept the "white" parts of the image at 255 (but strangely, the "replace" default value is just 1!), and set the definition for the dark parts of the image to different values (ranging from about 10 to about 240 in steps of 10) based on the For loop index.
 - c. While still in the For loop, I used "IMAQ Particle Analysis" on each thresholded image. Specifically, I requested data about x, y, and area for each "particle".
 - d. I evaluated each thresholded image using the particle data. Specifically, a thresholded image was OK if it had at least two particles (being the top and bottom of the string!), and the two largest particles accounted for at least 90% of the area of all the particles found, and each of the two largest particles was smaller than 2% of the area of the entire image. Usually, more than one thresholded version of the image was able to meet these criteria; in those cases, I used the most aggressive threshold (but I don't think it matters much).
 - e. Now that I had (x, y) coordinates for the two largest particles, the one that is higher vertically is the "pivot". I then used the Pythagorean theorem and arctangent to compute r and θ for each image. My value of r was just in pixels, because neither the period nor damping coefficients are dependent on the physical scale.
5. The non-linear curve fit tool is (surprisingly) not case sensitive. This means that you can't have separate variables called both " t " and " T ". My fitting formula for $\theta(t)$ had amplitude, phase, period, damping coefficient, and a vertical offset (in case the camera wasn't perfectly vertical). You might look up "damped sine waves" anywhere on the internet first, and just add the vertical offset to then end of the formula.
6. The non-linear curve fit requires initial guesses for each parameter.
 - a. I used the "length" input to determine an initial guess for the period using "physics" (it's the only reason the length input is even there).
 - b. My guess for the angular amplitude was $\frac{1}{2}(\theta_{\max} - \theta_{\min})$ (note that I expected θ_{\min} to be negative in this formula!).
 - c. My guess for the damping coefficient was just 5 seconds. I expect that it will actually be in the hundreds of seconds, but the non-linear curve fit gave better results when it was initialized too small. If the damping coefficient came out to be negative, I used zero instead. This happened whenever the orientation of the pendulum moved significantly out-of-plane due to precession.
 - d. My initial guess for the vertical offset was $\frac{1}{2}(\theta_{\max} + \theta_{\min})$ (note again that I expected θ_{\min} to be negative in this formula!). Again, this value became more significant when the motion was out-of-plane. It should be near zero in any case, if your camera isn't tilted to the side.
 - e. I used a finite-difference method to compute the initial slope of $\theta(t)$, and then used this to help make a good estimate of the initial phase. Specifically, I used Mathematica to learn that:

$$\phi = -\sin^{-1}\left(\frac{[\text{Initial Slope}] \cdot \text{Period}}{2\pi \cdot \text{Amplitude}}\right)$$

The results from this formula were mediocre, but apparently worked well enough. This formula assumes that $\theta(t)$ is in the form of a cosine, not a sine.