Synchronizované svíčky

Fyzikální seminář 2021, zimní semestr

vypracovali G. Fojtová, J. Trojan, O. Vala a M. Indra

22. října 2021

Fakulta jaderná a fyzikálně inženýrská v Praze



Motivace



Obrázek 1: Svíčky v Katedrále svatého Petra, Kolín nad Rýnem, Německo [1]

- prozkoumat jev provázaných oscilujících svíček
- provézt pozorování jevu a porovnat výsledky se studiemi
- stanovit průměrné frekvence oscilací z videozáznamu
- porovnání naměřených frekvencí pro různé vzdálenosti oscilátorů

Obsah

- úvod do problematiky
- teorie
- metodika
- výsledky pozorování
- závěr a diskuze

Teorie

Hoření



Obrázek 2: Oscilátory použité v našem experimentu

Podle studie z roku 2019 [5]:

- Jedna oscilátor nevykazuje oscilační chování, u dvou lze pozorovat menší změny jasu a amplitudy plamenů.
- Výrazné oscilace byly pozorovány pro 3 a více svíček pevně vázaných do jednoho oscilátoru.

Podle studie z roku 2016 [2]:

- Skládaní oscilátorů závisí na mnoha faktorech.
- Existuje několik nestabilních režimů oscilace.
- Počet oscilátorů mění chovaní celého systému.
- Jas oscilátorů se mění v těchto systémech asymetricky.
- Pokud je systém z více než 3 oscilátorů může docházet k přibližování a oddalování plamenů.

Podle studie z roku 2009 [7]:

- Dle studie lze popsat jev modelem.
- Tento model je složen ze dvou obyčejných diferenciálních rovnic, z nichž jedna popisuje změnu teploty a druhá změnu koncentrace kyslíku.
- Z výzkumu vyplývá, že má existovat kritická vzdálenost mezi svíčkami
- Pokud je tato vzdálenost překročena svíčky začnou oscilovat v protifázi.

Metodika

Konstrukce oscilátorů





Obrázek 3: Ukázka použitých oscilátorů

Obrázek 4: Balení použitých svíček

Knoty svíček tvoří rovnostranný trojúhélník o straně
 d = (8 ± 1) mm.

Příprava experimentu

Použité nástroje

- fotoaparát, Redmi T8 (240 FPS, 720P) + stojan
- posuvné meřidlo

Použité materiály

- polystyrenový blok
- schéma rozložení (Autodesk AutoCAD)
- polypropylenový box
- alobal, lepicí páska a páratka

Schéma rozložení



Obrázek 5: Vytisknuté schéma připevněné k polystyrenovému bloku

Experimentální aparatura



Obrázek 6: Aparatura použitá pro provedení experimentu

Postup měření

- 7. března 2021
- měření trvalo 3 hodiny (od 17.00 do 20.00)
- teplota v místnosti byla v průběhu měření konstantní (t = 18,8 °C).

Postup měření byl následující

- 1. oscilátory byly umístěny na vyznačené pozice
- 2. knoty svíček byly zkráceny na stejné délky
- 3. všechna světla v místnosti byla ztlumena
- 4. oscilátory byly zapáleny (všechny knoty najednou)
- 5. bylo zapnuto natáčení
- 6. mezi každým měřením pauza 10 minut

Analýza dat

Dvě navržené metody získání frekvence oscilátorů

- odhad krátkých intervalů pomocí rychlé Fourierovy transformace (FFT)
- výpočet kořenů interpolace hrubou silou

Využitý software

- Blender 2.91.2 (editování a přerenderování videí)
- Python 3.9 (numpy, cv2, PIL, pandas, scipy.fftpack)

Vizualizace dat

- GNUPLOT 5.4
- Python 3.9 (matplotlib.pyplot)

Otsu metoda (thresholding)

Námi upravená metoda zpracování obrazu [5]



Obrázek 7: Ilustrační sekvence obrázků zobrazující cyklus jednoho oscilátoru

Ukázka algoritmu



Obrázek 8: Youtube video s QR kód odkazem na ukázku https://www.youtube.com/watch?v=YfmwdZsaVNE

Analýza dat metoda 1



Aproximace pomocí FFT



Obrázek 10: propojené oscilátory ve fázi, jejich odpovídající aproximace a frekvenční 17/33 spektra

Analýza dat metoda 2

- tzv. "bruteforce" řešení
- zpracování Otsu metodou, normalizace dat
- kubická interpolace naměřených bodů dat
- hledání průniků interpolace s průměrnou hodnotou na daném intervalu dat

Ukázka druhého řešení



Obrázek 11: vizualizace výsledků popsané metody

Výsledky pozorování

Ve fázi



Obrázek 12: Graf oscilací ve fázi pro vzdálenost 30 mm

V protifázi



Obrázek 13: Graf oscilací v protifázi pro vzdálenost 50 mm

Totální dominance



Obrázek 14: Graf totální dominance pravého oscilátoru pro vzdálenost 110 mm 22/33

Částečná dominance 1



Obrázek 15: Graf částečné dominance pravého oscilátoru pro vzdálenost 90 mm 23/33

Částečná dominance 2



Obrázek 16: Graf částečné dominance levého oscilátoru pro vzdálenost 50 mm 24/33

Změna dominance



Obrázek 17: Graf změny vzájemné dominance oscilátorů ve vzdálenosti 90 mm 25/33

Death mode



Obrázek 18: Smrt provázaných oscilátorů, tzv death mode

Výsledky

Výsledky pro metodu 1

Vzdálenost	Levý oscilátor	Pravý oscilátor
<i>di</i> (mm)	$f_{li} = (\overline{x_{li}} \pm \sigma_{li}) \text{ Hz}$	$f_{ri} = (\overline{x_{ri}} \pm \sigma_{ri}) \text{ Hz}$
30	10,4 \pm 0,5	10,4 \pm 0,5
50	11,8 \pm 0,7	$9,9\pm0,4$
70	-	-
90	11,0 \pm 0,5	$11,2\pm0,3$
110	10,8 \pm 0,6	10,8 \pm 0,6
130	11,0 \pm 0,3	$9,9\pm0,1$
150	11,0 \pm 0,6	10,5 \pm 0,6
170	11,6 \pm 0,5	10,5 \pm 0,6

Tabulka 1: Pomocí FFT vypočtené průměrné frekvence pro jednotlivé vzdálenosti

Výsledky pro metodu 2

Tabulka 2: Pomocí bruteforce metody vypočtené průměrné frekvence pro jednotlivé vzdálenosti

Vzdálenost	Levý oscilátor	Pravý oscilátor
<i>di</i> (mm)	$f_{li} = (\overline{x_{li}} \pm \sigma_{li}) \text{ Hz}$	$f_{ri} = (\overline{x_{ri}} \pm \sigma_{ri}) \text{ Hz}$
30	$10,23\pm0,08$	10,4 \pm 0,2
50	$11,90\pm0,10$	10,5 \pm 0,1
70	-	-
90	$12{,}50\pm0{,}10$	11,2 \pm 0,3
110	10,3 \pm 0,5	10,4 \pm 0,2
130	11,07 \pm 0,03	11,50 \pm 0,01
150	10,4 \pm 0,4	$11.0\pm0,2$
170	$11.0\pm0,3$	$10,92\pm0,07$

Porovnání výsledků



Obrázek 19: Graf pozorovaných frekvencí v závislost na vzdálenosti pro levý oscilátor

Porovnání výsledků



Obrázek 20: Graf pozorovaných frekvencí v závislost na vzdálenosti pro pravý oscilátor

Shrnutí pozorování

Tabulka 3: Shrnutí všech pozorovaných oscilačních módů pro jednotlivé vzdálenosti

$d_i (mm)$	Pozorované módy
30	ve fázi, malý fázový posun, smrt
50	ve fázi, v protifázi, částečná dominance
70	-
90	v protifázi, změna dominance, smrt
110	v protifázi, totální dominance
130	v protifázi, částečná dominance
150	v protifázi, totální dominance
170	v protifázi, změna dominance, smrt

- pozorovány všechny popsané módy oscilací pro systém dvou oscilátorů viz tabulka 3
- z grafů na obrázcích 19, 20 lze vypozorovat změnu frekvence v různých vzdálenostech
- z měření nelze jednoznačně stanovit kritická vzdálenost předpovídaná studiemi (náš odhad 50 mm - 90 mm)

Chyby měření

- FFT, zkreslené spektrum, aliasing, chyba únikem [3]
- chyba interpolace
- vnější proudění vzduchu (při zapalování)
- nízká snímkovací frekvence (240 FPS)
- malý zkušební vzorek dat (pro chaotický jev)
- nečistoty v parafínu, složení [2]

Chyby byly kompenzovány použitými metodami.

Reference

- BORMANS, Thomas. Closeup photo of lighted tealight candles on rack. Unsplash: The internet's source of freely-usable images. [online]. Katedrála svatého Petra, Kolín nad Rýnem, Německo, 31. 9. 2017 [cit. 2021-10-16]. Dostupné z: tinyurl.com/e2tavsb7
- [2] Okamoto, K. et al. Synchronization in flickering of three-coupled candle flames. Sci. Rep. 6, 36145; doi: 10.1038/srep36145 (2016).

Reference

- [3] FFT, Katedra Fyziky, PřF UK, [online]. Copyright © [cit. 12.03.2021]. Dostupné z: http://physics.ujep.cz/ ~mmaly/vyuka/poc_fyz_2/FT/EXPME_3.pdf
- [4] Forrester, D. M. Arrays of coupled chemical oscillators. Sci. Rep. 5, 16994; doi: 10.1038/srep16994 (2015).
- [5] OpenCV & Python The Otsu's Binarization for thresholding – Meccanismo Complesso. Meccanismo Complesso – Mai nella stessa forma [online]. Copyright © 2021 Meccanismo Complesso [cit. 12.03.2021]. Dostupné z: https://tinyurl.com/fctkkc9j

Reference

- [6] Kitahata, H. et al. Oscillation and synchronization in the combustion of candles. The Journal of Physical Chemistry A 113, 8164–8168 (2009).
- [7] Oscillation and Synchronization in the Combustion of Candles, J. Phys. Chem. A 2009, 113, 8164-8168
 [online] Dostupné z: https://tinyurl.com/76y8ry2h

Prostor k diskuzi

Změna dominance



Obrázek 21: Změna frekvence v průběhu změny dominance oscilátorů

FFT



FFT



FFT



```
5 def _get_periods(roots):
 6
        if len(roots) % 2 == 0:
 7
            return [(roots[i+2]-roots[i]) for i in range(len(roots)-2)]
 8
        else:
 9
            return [(roots[i+2]-roots[i]) for i in range(len(roots)-3)]
    def get period(roots):
11
            periods = get periods(roots)
12
            return np.mean(periods), np.std(periods)/np.sqrt(len(periods))
13
   def get freq(roots):
14
            T, stdT = get period(roots)
15
            return (1/T, 1/(T^{**2})^{*}stdT)
16
17
18
   fl = interp1d(t, ndfl, kind='cubic', fill value="extrapolate")
    fr = interp1d(t, ndfr, kind='cubic', fill value="extrapolate")
19
20
21 rr = RootFinder(0, t[-1], dt)
22 rl = RootFinder(0, t[-1], dt)
23 roots right = np.sort(rr.find(fr))
24 roots left = np.sort(rl.find(fl))
```

```
1
   def get values(vid Left, vid right, treshold):
 2
3
        def get raw(vid, treshold):
           raw = []
 4
           cap = cv2.VideoCapture(str(vid))
 5
           if (cap.isOpened()== True): #check if it opens
 6
                while (cap.isOpened()):
 7
                    ret. frame = cap.read()
 8
                    if frame is None:
 9
                        break
10
                    frame = cv2.cvtColor(frame, cv2.COLOR BGR2GRAY)
11
                    frame = cv2.medianBlur(frame, 5)
12
                    ret, frame = cv2.threshold(frame, treshold, 255, cv2.THRESH BINARY)
13
                    raw.append(ImageStat.Stat(
14
                        Image.fromarrav(
15
                            np.uint8(cm.gist_earth(frame)*255))).rms[0])
16
           cap.release()
17
           return raw
18
        return _get_raw(vid_left, treshold), _get_raw(vid_right, treshold)
```

```
1 def normalize(raw_left, raw_right, START, END):

2 if START < 0:

    return -1

4 if (END > len(raw_left)) or (END > len(raw_right)):

6 norm_left = (raw_left[START:END] - np.mean(raw_left[START:END]))/np.max(raw_left[START:END])

7 norm_right = (raw_right[START:END] - np.mean(raw_right[START:END]))/np.max(raw_right[START:END])

8 return norm_left:-1].
```

```
class RootFinder:
        def init (self, start, stop, step=0.01, root dtype="float64", xtol=1e-9):
            self.start = start
           self.stop = stop
           self.step = step
            self.xtol = xtol
 7
            self.roots = np.array([], dtype=root dtype)
 8
 9
       def add to roots(self, x):
10
            if (x < self.start) or (x > self.stop):
                return # outside range
            if any(abs(self.roots - x) < self.xtol):</pre>
                return # root already found.
14
            self.roots = np.append(self.roots, x)
16
17
        def find(self, f, *args):
18
            current = self.start
19
20
            for x0 in np.arange(self.start, self.stop + self.step, self.step):
21
                if x\theta < current:
22
               x = self.find root(f, x0, *args)
24
               if x is None: # no root found.
25
                    continue
26
                current = x
27
                self.add to roots(x)
28
29
            return self.roots
30
31
        def find root(self, f, x0, *args):
32
            x, _, ier, _ = fsolve(f, x0=x0, args=args, full_output=True, xtol=self.xtol)
33
            if ier == 1:
34
                return x[0]
35
            return None
```