Линеаризация системы с управлением при помощи библиотеки SymPy

Велиева Т. Р.

Российский университет дружбы народов

«Компьютерная алгебра»,
ВМК МГУ, 12 апреля 2017 г.

Содержание

Цель

Автоматизация вычислений, направленных на линеаризацию модели.

Задачи

- Выбор средств компьютерной алгебры;
- Применение библиотеки SymPy к задаче автоматизации расчетов.

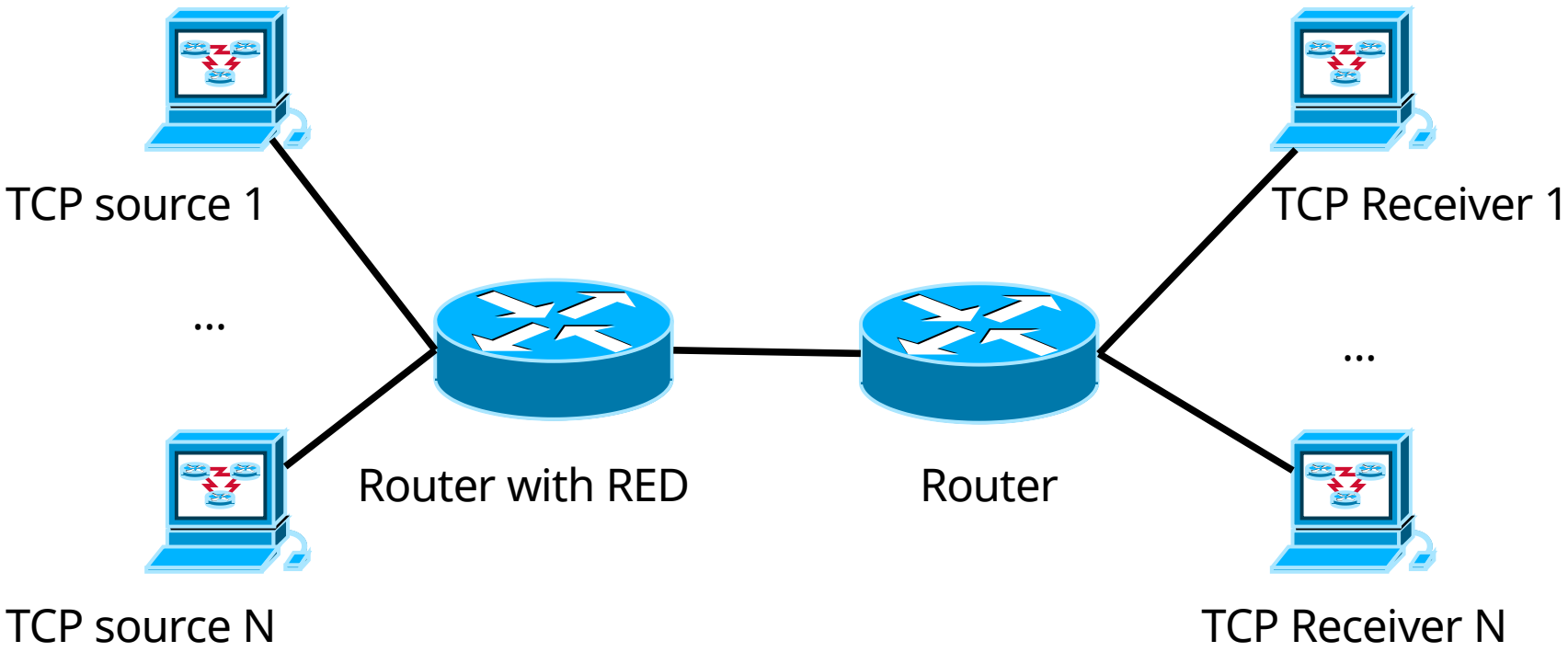

Рис. 1. Топология сети

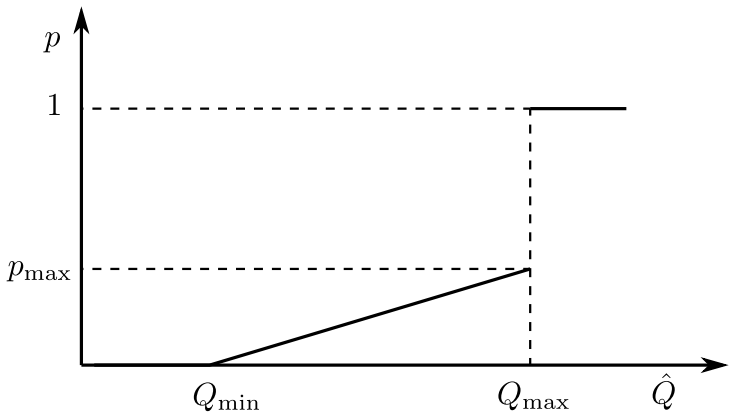

Рис. 2. Функция сброса RED

$$p(\hat{Q}) = \begin{cases} 0, & 0 < \hat{Q} \leqslant Q_{\min}, \\ \dfrac{\hat{Q} - Q_{\min}}{Q_{\max} - Q_{\min}} p_{\max}, & Q_{\min} < \hat{Q} \leqslant Q_{\max}, \\ 1, & \hat{Q} > Q_{\max}. \end{cases}$$

Методика гармонической линеаризации модели системы с управлением

- Линеаризуем исходную модель;
- Выделяем линейную часть;
- Вычисляем коэффициенты гармонической линеаризации;
- Применяем различные критерии для определения существования автоколебаний и их параметров.

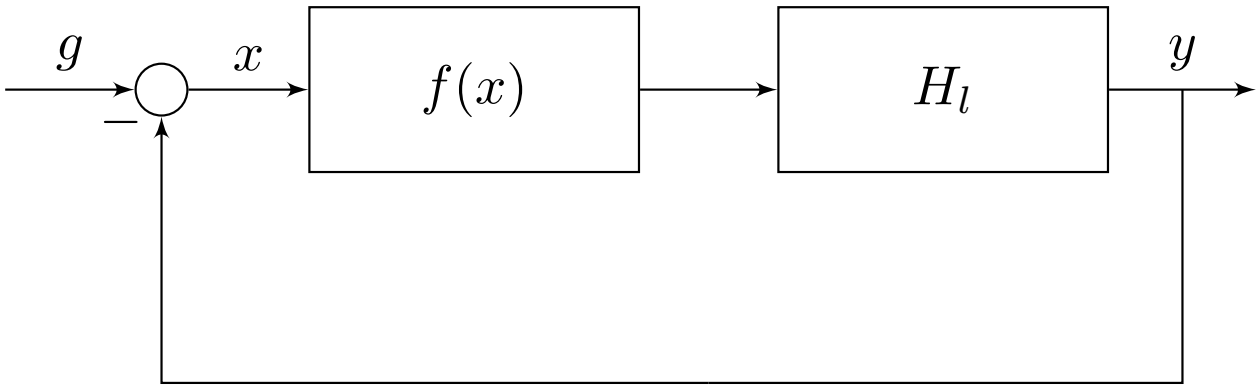

Рис. 3. Блочная структура системы для метода гармонической линеаризации

Приближенная передаточная функция нелинейного звена

$$f(x) = \left[\varkappa(A) + \frac{\varkappa'(A)}{\omega}\frac{\mathrm{d}}{\mathrm{d}t}\right]x = H_{nl}(A, \partial_t)x,$$

Коэффициенты гармонической линеаризации

$$\varkappa(A) = \frac{a_1}{A} = \frac{1}{A\pi}\int_0^{2\pi} f(A\sin(\omega t))\sin(\omega t)\,\mathrm{d}(\omega t)\,;$$

$$\varkappa'(A) = \frac{b_1}{A} = \frac{1}{A\pi}\int_0^{2\pi} f(A\sin(\omega t))\cos(\omega t)\,\mathrm{d}(\omega t)\,.$$

Критерий Найквиста–Михайлова I

Характеристическая функция системы

$$1 + H_o(\mathrm{i}\omega) = 0,$$
$$H_o(\mathrm{i}\omega) := H_l(\mathrm{i}\omega) H_{nl}(A, \mathrm{i}\omega).$$

H_o — передаточная функция разомкнутой системы.

Критерий Найквиста–Михайлова II

Метод Гольдфарба

$$H_l(\mathrm{i}\omega) = -\frac{1}{\varkappa(A) + \mathrm{i}\varkappa'(A)}.$$

Метод Коченбургера

$$\varkappa(A) + \mathrm{i}\varkappa'(A) = -\frac{1}{H_l(\mathrm{i}\omega)}.$$

Критерий Рауса–Гурвица I

Передаточная функция замкнутой системы

$$H_c(s) = \frac{H_l(s)}{1 + H_l(s)H_{nl}(s)} := \frac{P_n(s)}{P_d(s)}.$$

Характеристическое уравнением системы

$$P_d(s) = 0,$$

$$P_d(s) := a_0 s^n + a_1 s^{n-1} + \cdots + a_n.$$

Критерий Рауса–Гурвица II

Определитель Гурвица

$$\Delta = \begin{vmatrix} a_1 & a_3 & a_5 & \dots & 0 \\ a_0 & a_2 & a_4 & \dots & 0 \\ 0 & a_1 & a_3 & \dots & 0 \\ 0 & a_0 & a_2 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & a_n \end{vmatrix}.$$

Критерий Гурвица

Устойчивость все n главных диагональных положительны, $a_0 > 0$.

Апериодическая устойчивость $a_n = 0$.

Колебательная устойчивость $\Delta_{n-1} = 0$.

Точка равновесия

$$\begin{cases} 0 = \dfrac{1}{T_f} - \dfrac{W_f^2}{2T_f} p_f; \\ 0 = \dfrac{W_f}{T_f} N_f - C; \\ 0 = -w_q C \hat{Q}_f + w_q C Q_f. \end{cases}$$

Уравнения связи

$$\begin{cases} p_f = \dfrac{2}{W_f^2}; \\ W_f = \dfrac{CT_f}{N_f}; \\ \hat{Q}_f = Q_f. \end{cases}$$

Правые части системы

$$\begin{cases} L_W(W, W_T, Q, p) = \dfrac{1}{T} - \dfrac{WW_T}{2T}p; \\ L_Q(W, Q) = \dfrac{W}{T}N - C; \\ L_{\hat{Q}}(\hat{Q}, Q) = -w_q C\hat{Q} + w_q CQ. \end{cases}$$

Переменные: $W := W(t)$, $W_T := W(t-T)$, $Q := Q(t)$, $p := p(t-T)$.

Линеаризованная система

$$\begin{cases} \delta W(s) = -\dfrac{1}{s + \frac{N}{CT_f^2}\left(1 + \mathrm{e}^{-sT_f}\right)} \dfrac{C^2 T_f}{2N^2} \mathrm{e}^{-sT_f}\, \delta p(s)\,; \\ \delta Q(s) = \dfrac{1}{s + \frac{1}{T_f}} \dfrac{N}{T_f}\, \delta W(s)\,; \\ \delta \hat{Q}(s) = \dfrac{1}{1 + \frac{s}{w_q C}}\, \delta Q(s)\,. \end{cases}$$

Линеаризованная функция сброса

$$\delta p(s) = P_{\mathrm{RED}} \frac{1}{1 + \frac{s}{w_q C}} \delta Q(s) \,.$$

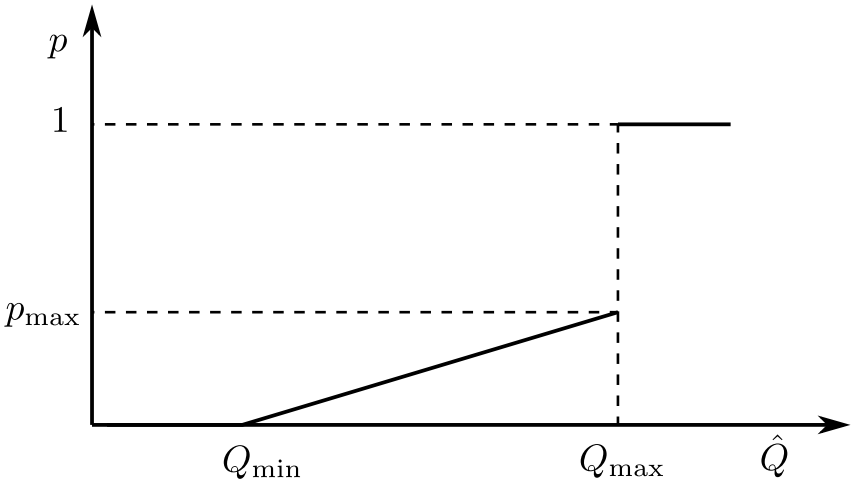

Рис. 4. Исходная функция сброса p

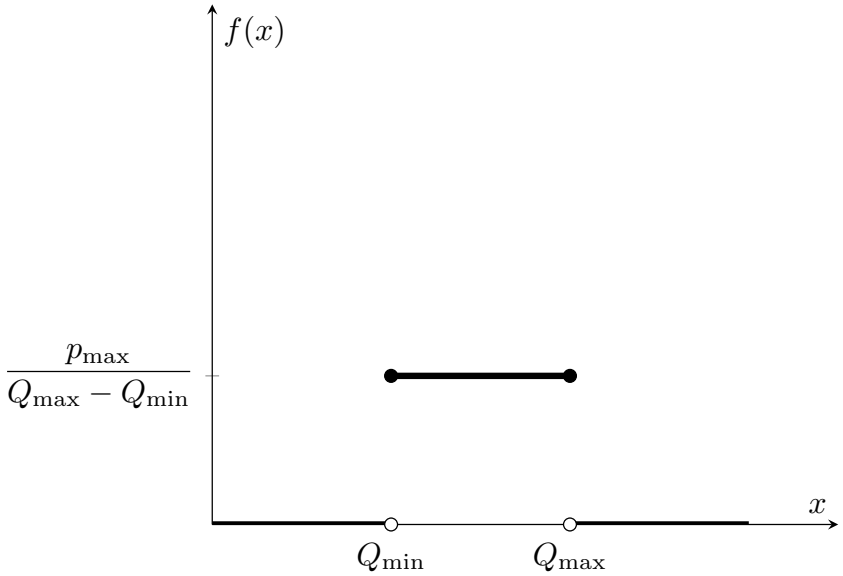

Рис. 5. Функция P_{RED}

Блочное представление линеаризованной модели RED

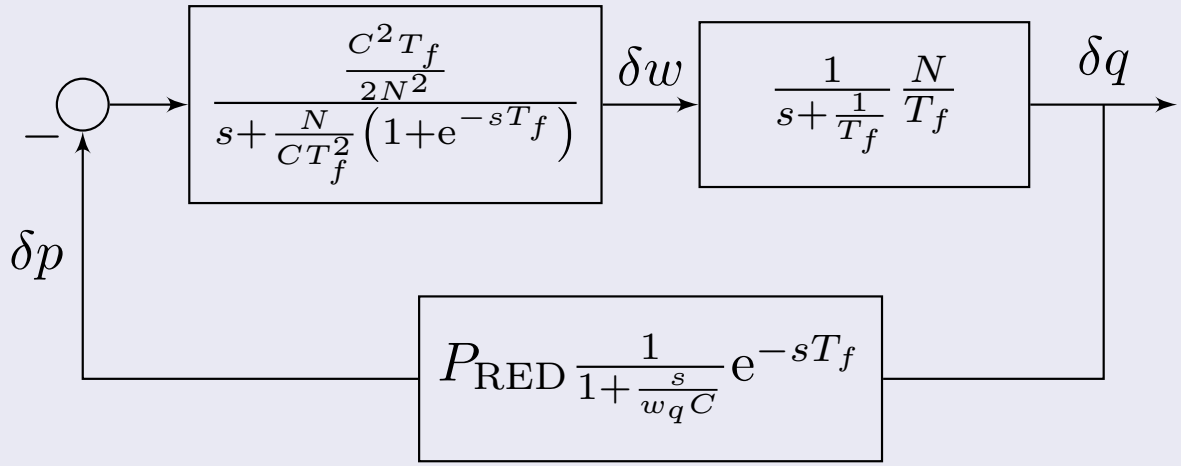

Рис. 6. Блочное представление линеаризованной модели RED

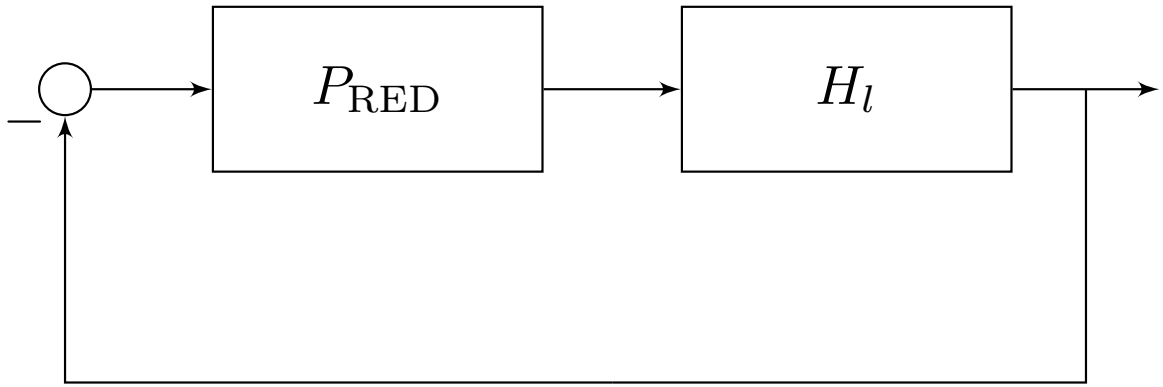

Рис. 7. Блочное представление линеаризованной модели RED для гармонической линеаризации

Уравнение Найквиста (вариант Гольдфарба)

$$\frac{1}{\mathrm{i}\omega + \frac{N}{CT_f^2}\left(1+\mathrm{e}^{-\mathrm{i}\omega T_f}\right)} \frac{1}{\mathrm{i}\omega + \frac{1}{T_f}} \frac{1}{1+\frac{\mathrm{i}\omega}{w_q C}} \frac{C^2}{2N} \mathrm{e}^{-\mathrm{i}\omega T_f} =$$

$$= -\frac{A\pi}{4p_{\max}} \left[\frac{1}{Q_{\max}-Q_{\min}} \left(\sqrt{1-\frac{Q_{\min}^2}{A^2}} - \sqrt{1-\frac{Q_{\max}^2}{A^2}} \right) + \mathrm{i}\frac{1}{A} \right]^{-1}.$$

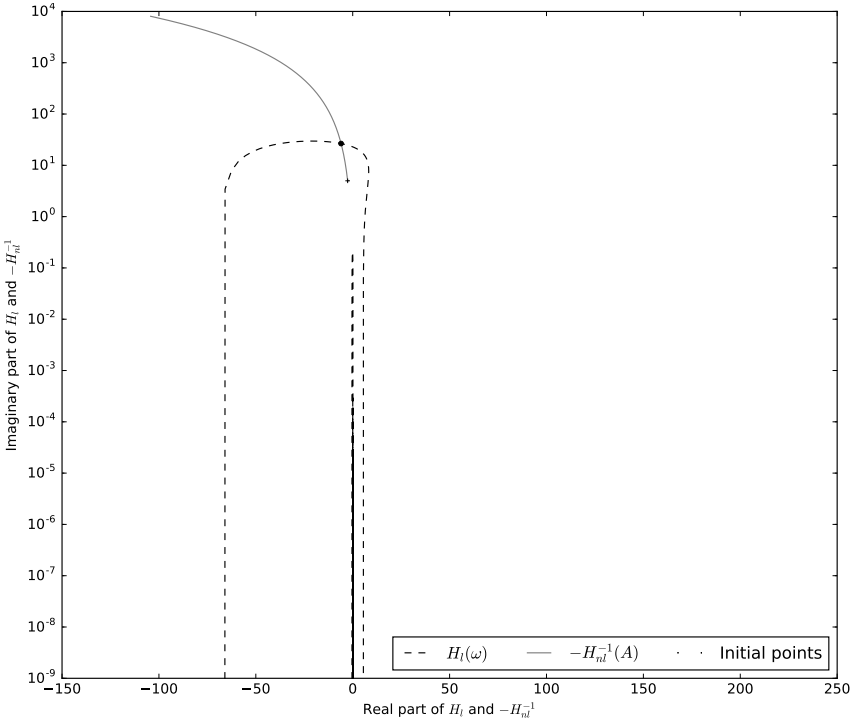

Рис. 8. Амплитудно-фазовая частотная характеристика (вариант Гольдфарба)

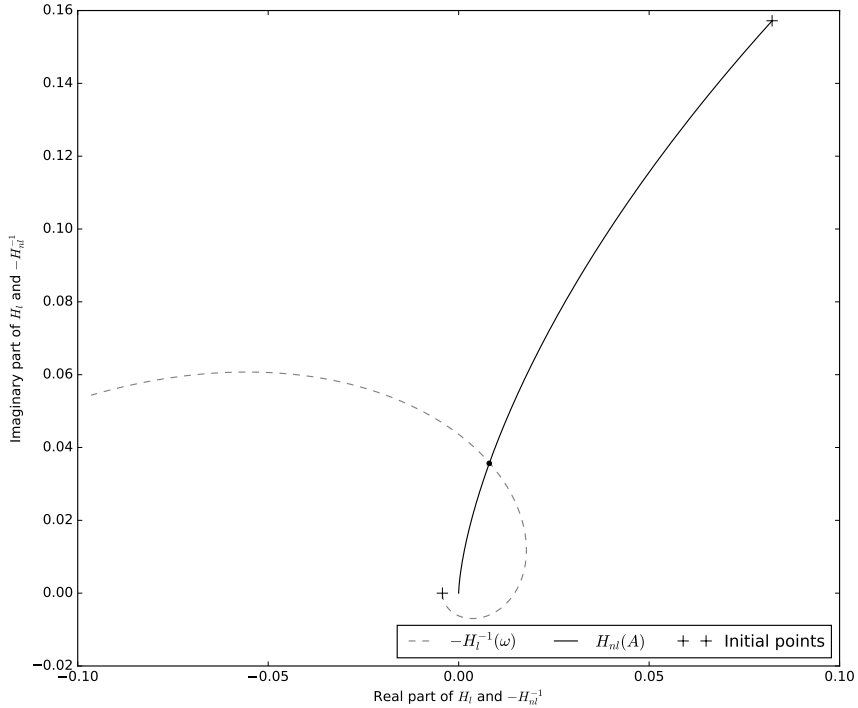

Рис. 9. Амплитудно-фазовая частотная характеристика для системы (вариант Коченбургера)

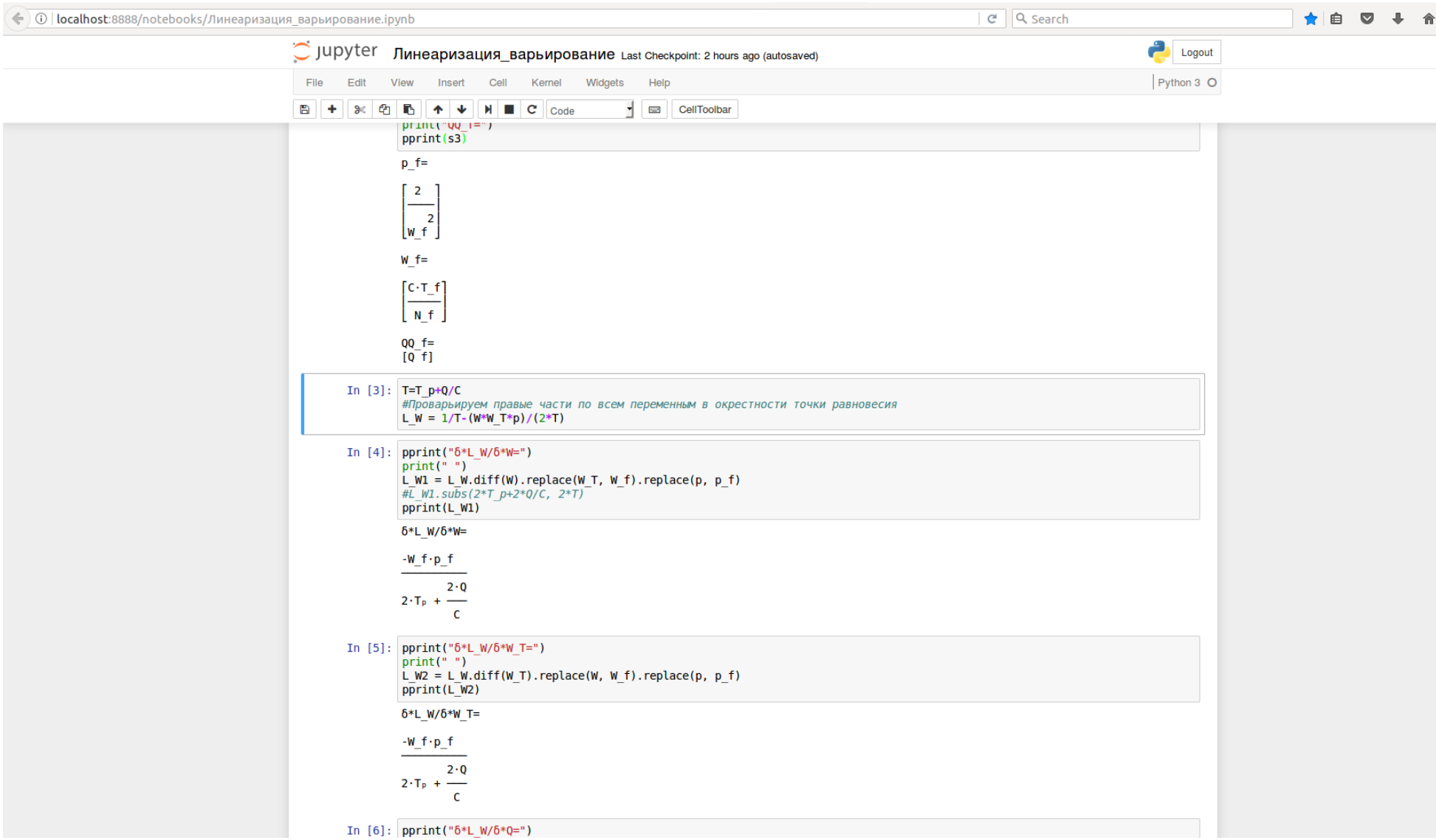

Рис. 10. Автоматизация расчетов при помощи Jupyter Notebook

```
#Задаем символьные переменные
p_f = Symbol("p_f")
W_f = Symbol("W_f")
QQ_f = Symbol("QQ_f")
Q_f = Symbol("Q_f")
N_f = Symbol("N_f")
C = Symbol("C")
T_f = Symbol("T_f")
T_p = Symbol("T_p")
Q = Symbol("Q")
QQ = Symbol ("QQ")
w_q = Symbol ("w_q")
W = Symbol("W")
W_T = Symbol("W_T")
p = Symbol("p")
T = Symbol("T")
```

```
N = Symbol("N")
L_W=Symbol("L_W")
L_Q=Symbol("L_Q")
L_QQ=Symbol("L_QQ")

#Получим уравнение связи на равновесные значения переменных
s1=solve(1/T_f-(W_f**2*p_f)/(2*T_f),p_f)
s2=solve((W_f/T_f)*N_f-C,W_f)
s3=solve(-w_q*C*QQ_f+w_q*C*Q_f,QQ_f)
pprint('p_f=')
pprint(s1)
print("W_f=")
pprint(s2)
print("QQ_f=")
pprint(s3)
```

```
#Варьирование правых частей функционала
# In[3]:
T=T_p+Q/C
L_W = 1/T-(W*W_T*p)/(2*T)

# In[4]:
pprint("δ*L_W/δ*W=")
L_W1 = L_W.diff(W).replace(W_T, W_f).replace(p, p_f)
#L_W1.subs(2*T_p+2*Q/C, 2*T)
pprint(L_W1)
# In[5]:
pprint("δ*L_W/δ*W_T=")
L_W2 = L_W.diff(W_T).replace(W, W_f).replace(p, p_f)
pprint(L_W2)
# In[6]:
pprint("δ*L_W/δ*Q=")
```

```
print(" ")
L_W3=L_W.diff(Q).replace(W, W_f).replace(W_T,W_f).replace(p, p
pprint(L_W3)
# In[7]:
pprint("δ*L_W/δ*p=")
L_W4=L_W.diff(p).replace(W, W_f).replace(W_T, W_f)
pprint(L_W4)
# In[8]:
L_Q=W/T*N-C
# In[9]:
pprint("δ*L_Q/δ*W=")
L_Q1=L_Q.diff(W)
pprint(L_Q1)
# In[10]:
pprint("δ*L_Q/δ*Q=")
L_Q2=L_Q.diff(Q).replace(W,W_f)
pprint(L_Q2)
```

```
# In[11]:
L_QQ=-w_q*C*QQ+w_q*C*Q
# In[12]:
pprint("δ*L_QQ/δ*QQ=")
L_QQ1=L_QQ.diff(QQ)
pprint(L_QQ1)
# In[13]:
pprint("δ*L_QQ/δ*Q=")
L_QQ2=L_QQ.diff(Q)
pprint(L_QQ2)
# In[14]:
W_f = C*T_f/N_f
pprint(W_f)
```