МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского» (ННГУ)

Институт информационных технологий, математики и механики

Кафедра: программной инженерии

Направление подготовки: «Фундаментальная информатика и информационные технологии»

ОТЧЕТ

по учебной практике

Тема:

«Оценка влияния запаздывания системы на сходимость ошибки алгоритма УИО»

Выполнил:

студент группы 381906-3

__________ Михеев И. А.

Подпись

Научный руководитель:

доктор физико-математических наук

__________ Пакшин П. В.

Подпись

г. Нижний Новгород, 2022 г.

Оглавление

Введение

Теория управления предоставляет множество методов для повышения качества процессов в динамической системе, но не всегда удается достичь желаемого результата, используя лишь традиционные методы. В частности, для нелинейных систем порой бывает очень нелегко достичь требуемой точности. Управление с итеративным обучением (УИО) (англ. Iterative Learning Control, ILC) является одним из сравнительно новых направлений теории управления. УИО можно классифицировать как одно из направлений интеллектуального управления для класса систем, которые работают в режиме повторений с фиксированным интервалом повторения. Благодаря воздействию на входной сигнал, итеративное обучение позволяет преодолеть недостатки традиционных методов, достигая высокого качества выполнения процессов.

Обобщенная цель алгоритма УИО состоит в изменении входного сигнала от одного повторения к следующему так, чтобы с увеличением числа повторении выходная переменная приближались к желаемой траектории. В отличие от *управления с обратной связью* и *адаптивного управления*, где используются лишь текущие значения переменных, характеризующих динамику процесса, процесс итеративного обучения состоит в коррекции закона управления на текущем повторении на основе информации, накопленной на предыдущем повторении, и информации о текущем состоянии системы.

Управление с итеративным обучением эффективно применяется к системам постоянно находящихся в повторяющемся режиме с предопределенным будущим поведением, что особенно характерно, например, для роботов-манипуляторов.

Постановка задачи

Рассмотрим модель с дискретным временем и со скалярными входными и выходными переменными с запаздыванием d вдоль отдельного повторения:

$$\begin{cases} x_k(t+1) = Ax_k(t) + Bu_k(t-d) \\ y_k(t) = Cx_k(t) \end{cases} \qquad (2.1)$$

где $x_k \in R^{n_u}$ – вектор состояния, $u_k \in R^{n_u}$ – вектор управления,

$y_k \in R^{n_y}$ – вектор выходных переменных, k – номер повторения,

d = 1÷3 – запаздывание вдоль отдельного повторения.

Также, введем в рассмотрение ошибку воспроизведения профиля:

$$e_k(t) = y_{ref}(t) - y_k(t),\ k = 0, 1, \ldots \qquad (2.2)$$

Задача состоит в нахождении такой последовательности управлений $u_k(t)$, которая обеспечивает достижение заданной точности воспроизведения желаемого результата за конечное число повторений k и оценке влияния запаздывания на сходимость ошибки.

Метод решения

Рассмотрим систему (2.1). Сформируем алгоритм управления с итеративным обучением для последовательности $u_k(t)$, $k = 1, 2, \ldots$, при заданном граничном условии (3.1.1):

$$u_{k+1}(t) = u_k(t) + \Delta u_{k+1}(t - d) \tag{3.1.1}$$

$$u_{k+1}(t) = u_k(t) + \Delta u_{k+1}(t) \tag{3.1.2}$$

Заметим, что отличие от общего алгоритма УИО (3.1.2), состоит в изменении Δu_{k+1} – корректирующей поправке к управлению на текущем шаге. То есть изменения в системе будут происходить с запаздыванием на d-итераций.

Ошибка обучения имеет вид:

$$e_k(t) = y_d(t) - y_k(t), \; k = 0, 1, \ldots \tag{3.2}$$

Также введем в рассмотрение приращения вектора состояния и вектора выходных переменных:

$$\xi_{k+1}(t + 1) = x_{k+1}(t) - x_k(t) \tag{3.3}$$

$$\eta_{k+1}(t + 1) = y_{k+1}(t) - y_k(t) \tag{3.4}$$

Динамика системы в терминах ошибки и приращения состояния описывается уравнениями:

$$\begin{aligned} &\xi_{k+1}(t + 1) = A\xi_{k+1}(t) + Bv_{k+1}(t), \; t = 0, 1, \ldots, N - 1 \\ &e_{k+1}(t) = - CA\xi_{k+1}(t) + e_k(t) - BA\xi_{k+1}(t), \; k = 0, 1, \ldots \\ &v_{k+1}(t) = \Delta u_{k+1}(t - d - 1) \end{aligned} \tag{3.5}$$

Зададим корректирующую поправку в виде:

$$\Delta u_{k+1}(t - d) = K_1\left(y_{k+1}(t) - y_k(t)\right) + K_2 e_k(t + 1) \tag{3.6}$$

При этом алгоритм УИО примет вид:

$$u_{k+1}(t) = u_k(t) + K_1\left(y_{k+1}(t) - y_k(t)\right) + K_2 e_k(t + 1) \tag{3.7}$$

Подставив (3.6) в (3.5), получим:

$$\xi_{k+1}(t\ +1) = (A + BK_1C)\xi_{k+1}(t) + BK_2e_k(t),$$

$$e_{k+1}(t) = - C(A + BK_1C)\xi_{k+1}(t) + (I - CBK_2)e_k(t), \qquad (3.8)$$

Введем обозначения:

$$\bar{\xi} = [\xi^T e^T],\ K = [K_1, K_2] \qquad (3.9)$$

$$\bar{A} = \begin{bmatrix} A & 0 \\ -CA & I \end{bmatrix},\ \bar{B} = \begin{bmatrix} B \\ -CB \end{bmatrix},\ \bar{C} = \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix}$$

Введем векторную функцию Ляпунова:

$$V\left(\xi_{k+1}(t), e_k(t)\right) = \begin{bmatrix} V_1\left(\xi_{k+1}\right)(t) \\ V_2\left(e_k(t)\right) \end{bmatrix} \qquad (3.10)$$

- $V_1\left(\xi_{k+1}(t)\right) > 0, \xi_{k+1} \neq 0,$
- $V_2(e_k(t)) > 0, e_k(t) \neq 0,$
- $V_1 = 0.$

и выберем ее компоненты в виде квадратичных форм:

$$V_1\left(\xi_{k+1}(t)\right) = \xi_{k+1}(t)^T P_1 \xi_{k+1}(t), \qquad (3.11)$$

$$V_2\left(e_k(t)\right) = e_k(t)^T P_2 e_k(t)$$

где $P_1, P_2 > 0$

Для сходимости ошибки потребуем выполнения неравенства:

$$DV\left(\xi_{k+1}(t), e_k(t)\right) = \bar{\xi}[(\bar{A} + \bar{B}K\bar{C})^T P(\bar{A} + \bar{B}K\bar{C}) - P + Q - C^{-T}K^T RK\bar{C} \geq 0$$

Которое эквивалентно:

$$P - (\bar{A} + \bar{B}K\bar{C})^T P(\bar{C} + \bar{B}K\bar{C}) - Q - \mathrm{C}^{-T}K^T RK\mathrm{C}^{-T} \geq 0 \qquad (3.13)$$

Билинейное матричное неравенство (3.13) будет выполнено, если будет разрешима относительно переменных X, Y, Z следующая система линейных и матричных неравенств и уравнений:

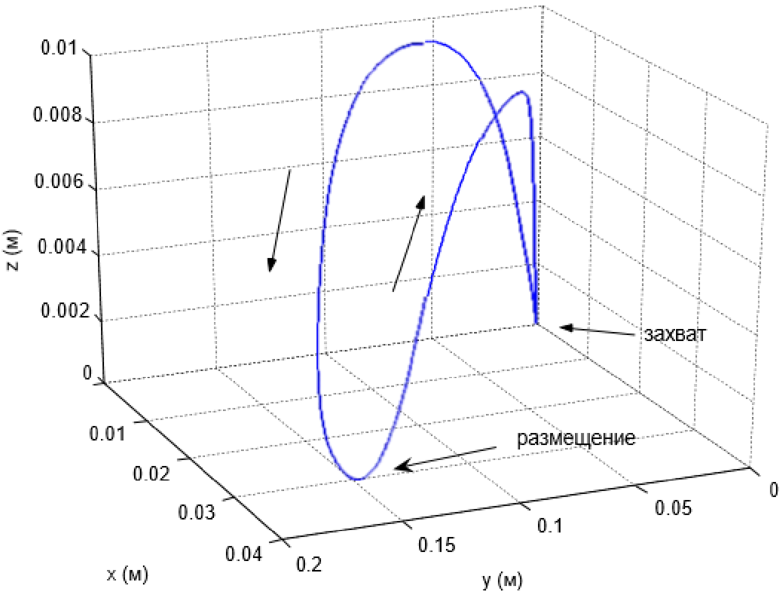
$$
\begin{bmatrix} X & (\bar{A}X + \bar{B}Y\bar{C})^{\mathrm{T}} & X & (Y\bar{C})^{\mathrm{T}} \\ \bar{A}X + \bar{B}Y\bar{C} & X & 0 & 0 \\ X & 0 & Q^{-1} & 0 \\ Y\bar{C} & 0 & 0 & R^{-1} \end{bmatrix} \succeq 0, \qquad (3.14)
$$

$$
\bar{C}X = Z\bar{C}
$$

На основе приведенных формул, можно сделать вывод, что введение запаздывания в корректирующей поправке к управлению на текущем шаге Δu_{k+1} кардинальным образом не изменит работу алгоритма УИО, но повлечет за собой увеличение числа повторений k, за которые будет достигнута заданная точность воспроизведения желаемого результата для последовательности управлений $u_k(t)$.

Полученные результаты

В качестве примера рассмотрим портальный робот из лаборатории университета Саутгемптона. Робот захватывает предметы из заданной области и устанавливает их с требуемой точностью на конвейер. Желаемая пространственная траектория движения представлена на рисунке:



Передаточная функция от управления к перемещению имеет вид:

$$G_y(S) = \frac{Y(S)}{U(S)} = \frac{23{,}7356(s+661{,}2)}{s(s^2+426{,}7s+1{,}744\bullet 10^5} \qquad (4.1)$$

Управление будем строить в рамках дискретной модели состояния. В рассматриваемом случае период дискретности $T_s = 0,01$ с. Параметры дискретной модели состояния получим с помощью стандартных функций ss и c2d библиотек Octave. Выбирая значения Q = diag[1 1 1 50] и R = 10 − 3, получим K = [−962,2 185,27]. В качестве меры точности воспроизведения желаемой траектории введем в рассмотрение среднеквадратическую ошибку обучения, которую будем вычислять по формуле:

$$E(k) = \sqrt{\frac{1}{N}\sum|e_k(t)|^2} \qquad (4.2)$$

График среднеквадратической ошибки обучения в зависимости от числа повторений, полученный без запаздывания вдоль повторения.

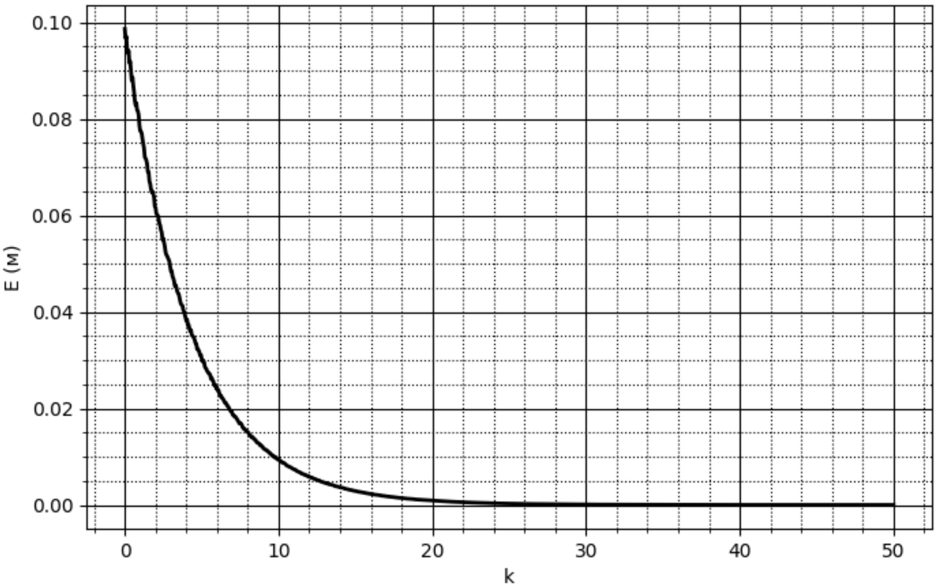


рис.1

Из графика (рис.1) видно, что уже после двадцати попыток вероятность ошибки сводится практически к нулю.

График среднеквадратической ошибки обучения в зависимости от числа повторений, полученный с запаздыванием вдоль повторения.

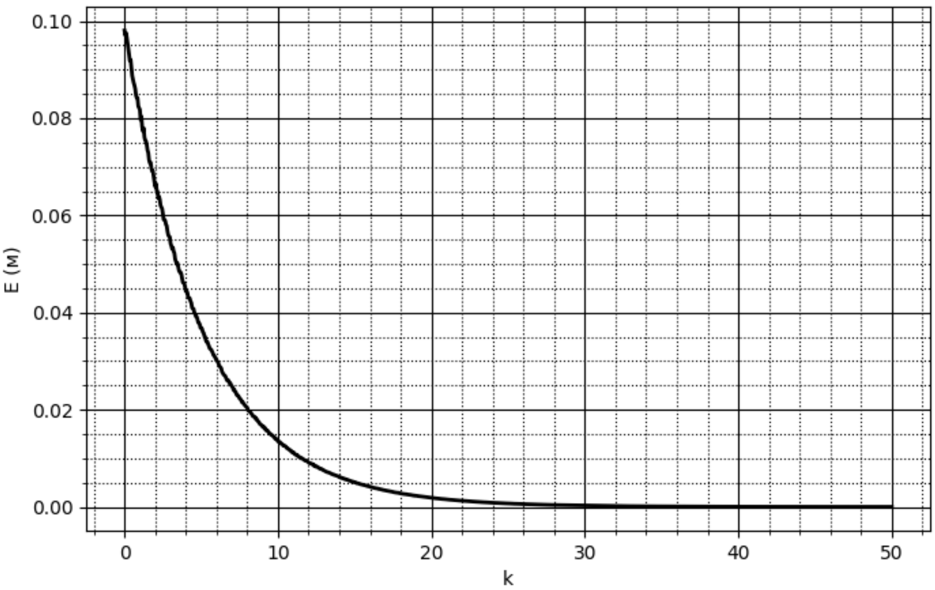


рис.2

Для случая с запаздыванием для сходимости ошибки значение числа попыток k получилось больше изначальных 20, увеличившись до 25, что негативно влияет на скорость достижения желаемой траектории.

Из полученных графиков можно сделать вывод, что добавление в систему запаздывания повлечет за собой увеличение числа повторений k, за которые будет достигнута заданная точность воспроизведения желаемого результата.

Заключение

В процессе выполнения лабораторной работы был теоретически описана схема выполнения алгоритма управления с итеративным обучения для системы с запаздыванием d вдоль отдельного повторения, выявлено влияние данного запаздывания на работу алгоритма и сходимость ошибки. Приведен и проанализирован пример использования алгоритма УИО с запаздыванием и без, представлены графически результаты работы алгоритма.

Список литературы

Литература:

1. Емельянова Ю.П., Пакшин П.В. Е 601 Матричные уравнения и неравенства в задачах теории управления: учеб. пособие / Ю.П. Емельянова, П.В. Пакшин; Нижегород. гос. техн. ун-т им. Р.Е. Алексеева. – Нижний Новгород, 2020.
2. Ю.В. Митришкин, А.Е. Коньков Метод линейных матричных неравенств в системах управления Учебное пособие. М.: Физический факультет МГУ, 2018.
3. А.Н. Чурилов, А.В. Гессен. (2004). Исследование линейных матричных неравенств. Путеводитель по программным пакетам. СПб.: Издательство Санкт-Петербургского университета, 2004.
4. DOUGLAS A. BRISTOW, MARINA THARAYIL, and ANDREW G. ALLEYNE A Survey of Iterative Learning // IEEE CONTROL SYSTEMS MAGAZINE » JUNE 2006.
5. AHN HYO-SUNG, CHEN YANG QUAN, MOORE K.L Iterative Learning Control: Brief Survey and Categorization // IEEE Trans. Syst., Man, Cybernet. Part C: Applications and Reviews. – 2007. – Vol. 37, No. 6. – P. 1099–1121.

Ресурсы сети Интернет:

1. Математическое описание систем автоматического управления https://habr.com/ru/post/520770/
2. Iterative Learning Control, Delays and Repetitive Control https://www.bristol.ac.uk/engineering/media/engineering-mathematics/anm-meetings/workshop-a/owens.pdf
3. Iterative Learning Control https://en.wikipedia.org/wiki/Iterative_learning_control
4. Ссылка на GitHub репозиторий с кодом программы: https://github.com/ITrickStar/ILC

Приложение

```
addpath(genpath('D:\packages\sedumi'));
addpath(genpath('D:\packages\YALMIP'));
Ts = 0.01;
Time = 1;
T = Time/Ts;
d = 3;
N = 30;
% параметры передаточной функции
y =@(s) 23.7356*(s+661.2);
x =@(s) s*(s^2+426.7*s+1.744*10^5);
X = zeros(T+1,1);
for t=d+1:T+1
   X(t,1) = x(t-d);
end
Y = zeros(T+1,1);
for t=1:T+1
   Y(t,1) = y(t);
end
% дискретная модель
sys = ss(x, y);
sysd = c2d(sys,Ts);
A = sysd.a;
B = sysd.b;
C = sysd.c;
I = eye(1);
A_=[ A; 0; -CA; I];
B_ = [B; -C*B];
C_ = [C, 0; 0, I];
Q = diag(1, 1, 1, 50);
R = 10-3;
K = [-962.2, 185.27];
% матричное неравенство
Cond11 = X;
```

```
Cond12 = (X*A_+B_*Y*C_)';
Cond13 = X;
Cond14 = (Y*C_)';
Cond21 = A_*X+B_*Y*C_;
Cond22 = X;
Cond31 = X;
Cond33 = inv(Q);
Cond41 = YC_;
Cond44 = inv(R);
Cond = [Cond11, Cond12, Cond13, Cond14; ...
       Cond21, Cond22, 0,      0; ...
       Cond31, 0,      Cond33, 0 ...
       Cond41, 0,      0,      Cond44];
options = sdpsettings('solver', 'sedumi', 'verbose');
solved = solvesdp(Cond >= 0, options);
% начальные значения
x0 = [0;0;0;0];
xhat0 = [0;0;0;0];
for k=1:N+1
   x{1,k}=x0;
   xhat{1,k}=xhat0;
   y(1,t)=C*x0;
end
for t=1:T+1
   u(t,1)=0;
   x{t+1,1}=A*x{t,1};
   xhat{t+1,1}=A*xhat{t,1};
   y(t,1)=C*x{t,1};
end
for k=2:N+1
   for t=1:T

u(t,k)=u(t,k-1)+K(1)*(xhat{t,k}-xhat{t,k-1})+K(2)*(yref(t+1,1)-C*x{t+1,k-
1});
       x{t+1,k}=A*x{t,k}+B*u(t,k);
       y(t,k)=C*x{t,k};
```

```
        xhat{t+1,k}=A*xhat{t,k}+B*u(t,k)-C*xhat{t,k};
        e(t,k-1)= Y(t,1)-y(t,k-1);
        E(1,k-1) = (E(1,k-1) + (e(t,k-1)^2));
    end
    E(1,k-1) = sqrt(E(1,k-1)/T); % среднеквадратическая ошибка обучения
end
% график
[XX, YY] = meshgrid(0:T-1,0:N-1);
figure(1);
plot(YY, E', 'b','LineWidth', 2);
title(str, 'FontSize',14);
xlabel('k','FontSize',14);
ylabel('E','FontSize',14);
grid on;
```