СРАВНИТЕЛЬНОЕ МОДЕЛИРОВАНИЕ СИСТЕМ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ КОНЦЕНТРАЦИЕЙ КРАСИТЕЛЯ ПРИ КРАШЕНИИ ТЕКСТИЛЬНЫХ МАТЕРИАЛОВ С ИСПОЛЬЗОВАНИЕМ ПИ-РЕГУЛЯТОРА И РЕГУЛЯТОРА НА ОСНОВЕ НЕЧЕТКОЙ ЛОГИКИ В MATLAB/SIMULINK

Ж.М.Муродов Н.А.Ниязова

Ташкентский институт текстильной и легкой промышленности

Аннотация. *В статье рассматривается задача автоматического управления концентрацией красителя в процессе крашения текстильных материалов. Для обеспечения требуемого качества окрашивания необходимо поддерживать концентрацию красителя на заданном уровне и своевременно компенсировать отклонения, возникающие в процессе обработки материала. Рассматриваются два метода управления: пропорционально-интегральный (ПИ) регулятор и регулятор на основе нечеткой логики. Приведена математическая модель объекта управления с учетом инерционности и транспортного запаздывания. Подробно рассмотрены формулы расчета ошибки регулирования, закона управления ПИ-регулятора, приращения управляющего сигнала нечеткого регулятора и показателей качества переходного процесса. Предложена структура реализации обеих систем в MATLAB/Simulink. Сравнение проводится по интегральной абсолютной ошибке, времени установления, перерегулированию и характеру управляющего воздействия.*

Ключевые слова: *текстильные материалы, крашение, концентрация красителя, автоматическое управление, ПИ-регулятор, нечеткая логика, математическое моделирование, MATLAB, Simulink, переходный процесс.*

Abstract. This article examines the problem of automatic control of dye concentration in the textile dyeing process. To ensure the required dyeing quality, the dye concentration must be maintained at a specified level, and deviations arising during material processing must be compensated for in a timely manner. Two control methods are considered: a proportional-integral (PI) controller and a fuzzy logic-based controller. A mathematical model of the control object is presented, taking into account its inertia and transport delay. The formulas for calculating the control error, the PI controller control law, the increment of the fuzzy controller's control signal, and the performance indices of the transient response are discussed in detail. The implementation structures of both systems in MATLAB/Simulink are proposed. The comparison is based on the integral absolute error, settling time, overshoot, and the characteristics of the control action.

Keywords: textile materials, dyeing, dye concentration, automatic control, PI controller, fuzzy logic, mathematical modeling, MATLAB, Simulink, transient response.

Annotatsiya. Maqolada to‘qimachilik materiallarini bo‘yash jarayonida bo‘yoq konsentratsiyasini avtomatik boshqarish masalasi ko‘rib chiqilgan. Bo‘yashning talab etilgan sifatini ta’minlash uchun bo‘yoq konsentratsiyasini belgilangan darajada saqlash va materialga ishlov berish jarayonida yuzaga keladigan og‘ishlarni o‘z vaqtida kompensatsiyalash zarur. Boshqarishning ikkita usuli: proporsional-integral (PI) rostlagich va noaniq mantiqqa asoslangan rostlagich ko‘rib chiqilgan. Boshqarish obyektining inersion xususiyatlari va transport kechikishini hisobga oluvchi matematik modeli keltirilgan. Rostlash xatosini hisoblash, PI-rostlagichning boshqarish qonuni, noaniq mantiqqa asoslangan rostlagich boshqaruv signalining orttirmasi hamda o‘tish jarayonining sifat ko‘rsatkichlarini aniqlash formulalari batafsil yoritilgan. Har ikkala tizimni MATLAB/Simulink muhitida amalga oshirish tuzilmalari taklif etilgan. Tizimlarni taqqoslash integral absolyut xato, rostlanish

vaqti, o‘ta rostlanish va boshqaruv ta’sirining xususiyatlari asosida amalga oshiriladi.

Kalit so‘zlar: to‘qimachilik materiallari, bo‘yash, bo‘yoq konsentratsiyasi, avtomatik boshqarish, PI-rostlagich, noaniq mantiq, matematik modellashtirish, MATLAB, Simulink, o‘tish jarayoni.

Введение. Крашение является одним из важнейших этапов текстильного производства. Качество готового материала зависит от концентрации красителя, температуры раствора, продолжительности обработки, интенсивности перемешивания и свойств текстильных волокон.

Если концентрация красителя изменяется относительно заданного значения, это может привести к неоднородности окрашивания и снижению качества продукции. Ручное управление не всегда позволяет своевременно компенсировать такие отклонения. Поэтому применение автоматических систем контроля и регулирования имеет большое значение для повышения стабильности технологического процесса.

Одним из распространенных способов управления является использование ПИ-регулятора. Он учитывает текущую ошибку и накопленное отклонение регулируемой величины. Однако при наличии транспортного запаздывания и изменении параметров объекта для получения требуемого качества регулирования может потребоваться дополнительная настройка.

Другим подходом является применение регулятора на основе нечеткой логики. Он формирует управляющий сигнал с использованием правил вида «если ошибка большая, то увеличить управляющее воздействие». Такой подход позволяет учитывать не только величину ошибки, но и направление её изменения.

Цель исследования — разработать и сравнить математические модели системы управления концентрацией красителя с ПИ-регулятором и регулятором на основе нечеткой логики в MATLAB/Simulink.

Для достижения цели необходимо: составить математическую модель объекта управления; записать закон управления ПИ-регулятора; разработать алгоритм нечеткого управления; реализовать обе системы в MATLAB/Simulink; определить показатели качества регулирования и сравнить результаты.

Теоретические исследования. В процессе крашения текстильных материалов важное значение имеет поддержание концентрации красильного раствора на заданном уровне. Для этого измеряется текущая концентрация раствора, и в зависимости от отклонения её значения от заданного регулируется работа насоса подачи красителя или положение регулирующего клапана.

Для математического описания свойств красильной ванны используется следующая передаточная функция:

$$G_0(s) = \frac{K_0}{T_0 s + 1}$$

Обозначения в данной формуле имеют следующий смысл:

- $G_o(s)$ — передаточная функция красильной ванны;
- K_o — коэффициент усиления объекта;
- T_o — постоянная времени объекта, с;
- s — оператор Лапласа.

Для моделирования примем следующие исходные значения:

Параметр	Значения	Описание
K_o	2,7	Коэффициент усиления объекта
T_o	126 s	Постоянная времени объекта
τ	15 s	Транспортное запаздывание

Ts	1 s	Шаг расчёта
t_{max}	1500 s	Продолжительность моделирования

Эти значения приняты условно для выполнения расчётов и не являются результатами измерений, полученными в реальных производственных условиях.

Без учёта запаздывания. Подставим в формулу значения K_o=2,7 и T_o=126, приведённые в таблице:

$$T_0 \frac{dy(t)}{dt} + y(t) = K_0 u(t).$$

В данной формуле:

- y(t) — выходная величина объекта, то есть показатель концентрации в модели;
- u(t) — управляющий сигнал, подаваемый на объект;
- dy(t)/dt — скорость изменения выходной величины во времени;
- T_o и K_o — параметры объекта.

Запишем уравнение относительно скорости изменения y(t):

$$\frac{dy(t)}{dt} = \frac{K_0 u(t) - y(t)}{T_0}$$

Теперь подставим значения K_o=2,7 и T_o=126:

$$\frac{dy(t)}{dt} = \frac{2{,}7u(t) - y(t)}{126}$$

Это уравнение означает, что выходная величина объекта зависит от разности между управляющим сигналом и текущим значением выходной величины. При изменении управляющего сигнала концентрация также изменяется с течением времени.

Для выполнения расчётов в MATLAB преобразуем это уравнение в дискретную форму:

$$y_{k+1} = y_k + \frac{T_s}{T_0}(K_0 u_k - y_k)$$

Здесь:

- y_k — выходная величина на текущем шаге;
- y_{k+1} — выходная величина на следующем шаге;
- u_k — управляющий сигнал на текущем шаге;
- T_s — шаг расчёта по времени.

Например, при T_s=1 с:

$$y_{k+1} = y_k + \frac{1}{126}(2{,}7u_k - y_k).$$

В запишем программу в Matlab: y(k+1) = y(k) + Ts/To(Ko*u(k)-y(k));*

Для приведения управляющего сигнала в соответствие с условным рабочим диапазоном насоса или клапана вводится следующее ограничение:

$$0 \le u_k \le 1$$

При этом принимается, что u_k=0 соответствует минимальному управляющему воздействию, а u_k=1 — максимальному управляющему сигналу.

Разработка ПИ-регулятора. ПИ-регулятор является одним из наиболее широко используемых алгоритмов управления в промышленных системах автоматического регулирования. Он состоит из пропорциональной и интегральной составляющих и формирует управляющий сигнал в зависимости от ошибки регулирования.

Ошибка регулирования определяется следующим образом:

$$e(t) = r(t) - y(t)$$

где r(t) — заданное значение, y(t) — выходная величина объекта.

Передаточная функция ПИ-регулятора:

$$G_{PI}(s) = K_p + \frac{K_i}{s}.$$

Закон управления в области времени:

$$u(t) = K_p e(t) + K_i \int_0^t e(\lambda) d\lambda.$$

В рассматриваемом расчётном примере выбираются следующие коэффициенты: Kp=1,94, Ki=0,014.

Данные коэффициенты приняты для предварительного моделирования. Их оптимальность необходимо проверить с учётом модели объекта, ограничений управляющего сигнала и результатов моделирования.

При дискретных вычислениях интегральная составляющая обновляется следующим образом:

$$I_k = I_{k-1} + K_i T_s e_k.$$

Управляющий сигнал:

$$u_k = \text{sat}(K_p e_k + I_k),$$

где (·) — оператор ограничения управляющего сигнала в заданном диапазоне.

В практических системах при ограничении управляющего сигнала необходимо предотвращать чрезмерное накопление интегральной составляющей. Это явление называется интегральным насыщением (integral windup). Поэтому в модели применяется механизм корректировки интегральной составляющей с учётом ограничений управляющего сигнала.

Разработка регулятора на основе нечёткой логики

Структура алгоритма управления**.** В регуляторе на основе нечёткой логики управляющее воздействие формируется не с помощью точного аналитического уравнения, а посредством лингвистических правил и функций принадлежности. Такой подход позволяет определять управляющее воздействие в зависимости от величины ошибки регулирования и тенденции её изменения.

В данном исследовании рассматривается нечёткая логическая система типа Мамдани с двумя входами и одним выходом.

Входные переменные:

• e_k — ошибка регулирования;

• $\Delta e_k = e_k - e_{k-1}$ — изменение ошибки регулирования.

Выходная переменная:

- Δu_k — изменение управляющего сигнала.

Управляющий сигнал обновляется следующим образом:

$$u_k = \text{sat}(u_{k-1} + K_u \Delta u_{F,k}),$$

где $\Delta u_{F,k}$ — выход нечёткой логической системы, K_u — коэффициент масштабирования управляющего воздействия.

Такой инкрементальный алгоритм позволяет постепенно изменять управляющий сигнал в зависимости от ошибки регулирования и её изменения, вместо резкого изменения управляющего воздействия за один шаг.

Ошибка регулирования, изменение ошибки и управляющее воздействие описываются пятью лингвистическими значениями:

Обозначение	**Английское название**	**Значение на русском языке**
NB	Negative Big	Большое отрицательное
NS	Negative Small	Малое отрицательное
ZE	Zero	Близкое к нулю
PS	Positive Small	Малое положительное
PB	Positive Big	Большое положительное

Для входных и выходной переменных используются треугольные функции принадлежности. Их исходный диапазон принимается равным [-1;1].

Нормированная ошибка регулирования:

$$e_{N,k} = sat_{[-1,1]}\left(\frac{e_k}{E_{\max}}\right).$$

Изменение ошибки регулирования:

$$\Delta e_{N,k} = sat_{[-1,1]}\left(\frac{e_k - e_{k-1}}{D_{\max}}\right).$$

Здесь E_{max} — масштаб нормирования ошибки регулирования, D_{max} — масштаб нормирования изменения ошибки.

В расчётном примере принимаются значения $E_{max}=1$ и $D_{max}=0{,}05$. Эти значения также являются условными параметрами. Для реального процесса их следует выбирать с учётом единиц измерения и диапазонов изменения сигналов.

Правила нечёткого управления

Работа регулятора на основе нечёткой логики определяется экспертными правилами. В таблице ниже приведён исходный набор правил управления.

Ошибка e/изменение Δe	**NB**	**NS**	**ZE**	**PS**	**PB**
NB	NB	NB	NB	NB	NS
NS	ZE	NS	NB	NS	NS
ZE	PS	ZE	NS	ZE	PS
PS	NS	ZE	PS	PS	PB
PB	ZE	PS	PB	PB	PB

Правила, представленные в таблице, отражают следующую логику управления:

- Если концентрация значительно ниже заданного значения, управляющий сигнал увеличивается.
- Если концентрация значительно выше заданного значения, управляющий сигнал уменьшается.
- Если ошибка регулирования уменьшается, управляющее воздействие корректируется более осторожно.



• При приближении ошибки к нулю резкие изменения управляющего сигнала сокращаются.

Приведённые правила являются первоначальными экспертными правилами. Их оптимальность не гарантируется заранее. Эффективность правил и функций принадлежности необходимо проверять на основе результатов моделирования.

Сравнительное моделирование в MATLAB

Для сравнения ПИ-регулятора и регулятора на основе нечёткой логики оба алгоритма подключаются к одной и той же математической модели объекта. Заданное значение, начальные условия, продолжительность моделирования и ограничения управляющего сигнала остаются одинаковыми.

В MATLAB для создания нечёткой логической системы можно использовать функции старого интерфейса Fuzzy Logic Toolbox: newfis, addvar, addmf, addrule и evalfis. В современных версиях MATLAB вместо них могут применяться новые функции, основанные на объектах. Это различие отражено в официальной документации [3].

Экспериментальные исследования. Программный код MATLAB

Следующий код рассчитывает отдельные выходные сигналы для ПИ-регулятора и регулятора на основе нечёткой логики, сравнивает результаты на одном графике и позволяет вычислить основные показатели качества регулирования.

```
%% 1. Параметры объекта и моделирования
Ko = 2.7;       % Коэффициент передачи объекта
To = 126;       % Постоянная времени объекта, с
Ts = 1;         % Шаг моделирования, с
Tsim = 1500;    % Общее время моделирования, с
t = 0:Ts:Tsim;
N = length(t);
```

```
r = ones(1,N);  % Заданное значение концентрации
%% Ограничения управляющего сигнала
umin = 0;       % Минимальное управляющее воздействие
umax = 1;       % Максимальное управляющее воздействие
%% 2. Параметры ПИ-регулятора
Kp = 1.94;      % Пропорциональный коэффициент
Ki = 0.014;     % Интегральный коэффициент
yPI = zeros(1,N);  % Выход объекта с ПИ-регулятором
uPI = zeros(1,N);  % Управляющий сигнал ПИ-регулятора
ePI = zeros(1,N);  % Ошибка регулирования
I = 0;             % Начальное значение интегральной составляющей
%% 3. Модель с ПИ-регулятором
for k = 1:N-1
    % Вычисление ошибки регулирования
    ePI(k) = r(k) - yPI(k);
    % Формирование управляющего сигнала
    v = Kp*ePI(k) + I;
    % Ограничение управляющего сигнала
    uPI(k) = min(max(v,umin),umax);
    % Условие уменьшения интегрального насыщения
    if (v == uPI(k)) || ...
       (v > umax && ePI(k) < 0) || ...
       (v < umin && ePI(k) > 0)
        I = I + Ki*Ts*ePI(k);
    end
    % Дискретная модель объекта управления
    yPI(k+1) = yPI(k) + ...
        Ts/To*(Ko*uPI(k)-yPI(k));
```

```
end
% Ошибка и управляющий сигнал в последней точке
ePI(N) = r(N) - yPI(N);
uPI(N) = uPI(N-1);
%% 4. Создание системы нечеткой логики Мамдани
fis = newfis('Dyeing_Fuzzy','mamdani');
% Определение входных и выходной переменных
fis = addvar(fis,'input','e',[-1 1]);
fis = addvar(fis,'input','de',[-1 1]);
fis = addvar(fis,'output','du',[-1 1]);
% Лингвистические термы:
% NB - отрицательное большое;
% NS - отрицательное малое;
% ZE - близкое к нулю;
% PS - положительное малое;
% PB - положительное большое.
names = {'NB','NS','ZE','PS','PB'};
% Параметры функций принадлежности
mf = [-1  -1  -0.5;
      -1  -0.5  0;
      -0.5  0  0.5;
       0   0.5  1;
       0.5  1   1];
% Создание функций принадлежности
for j = 1:5
    fis = addmf(fis,'input',1,names{j}, ...
        'trimf',mf(j,:));
    fis = addmf(fis,'input',2,names{j}, ...
```



```
        'trimf',mf(j,:));
    fis = addmf(fis,'output',1,names{j}, ...
        'trimf',mf(j,:));
end
%% Формирование базы нечетких правил
R = [1 1 1; 1 2 1; 1 3 1; 1 4 2; 1 5 3; 2 1 1; 2 2 2; 2 3 2;  2 4 3; 2 5 4; 3 1 2; 3 2 2; 3 3 3; 3 4 4; 3 5 4; 4 1 2; 4 2 3; 4 3 4; 4 4 4;  4 5 5; 5 1 3; 5 2 4; 5 3 5; 5 4 5; 5 5 5];
% Вес каждого правила равен 1.
% Для операции AND используется оператор min.
R = [R ones(25,1) ones(25,1)];
% Добавление правил в систему нечеткой логики
fis = addrule(fis,R);
%% 5. Модель с нечетким регулятором
yF = zeros(1,N);  % Выход объекта с нечетким регулятором
uF = zeros(1,N);  % Управляющий сигнал
eF = zeros(1,N);  % Ошибка регулирования
Ku = 0.03;        % Коэффициент масштабирования управления
Emax = 1;         % Максимальное значение ошибки для нормирования
DEmax = 0.05;     % Масштаб приращения ошибки
for k = 1:N-1
    % Вычисление ошибки регулирования
    eF(k) = r(k) - yF(k);
    % Вычисление приращения ошибки
    if k == 1
        de = 0;
    else
        de = eF(k) - eF(k-1);
```

```
    end
    % Нормирование ошибки
    en = min(max(eF(k)/Emax,-1),1);
    % Нормирование приращения ошибки
    den = min(max(de/DEmax,-1),1);
    % Расчет выходного сигнала нечеткого регулятора
    du = evalfis([en den],fis);
    % Инкрементальный закон управления
    uF(k) = min(max(uF(max(k-1,1)) ...
        + Ku*du,umin),umax);
    % Дискретная модель объекта управления
    yF(k+1) = yF(k) + ...
        Ts/To*(Ko*uF(k)-yF(k));
end
% Ошибка и управляющий сигнал в последней точке
eF(N) = r(N) - yF(N);
uF(N) = uF(N-1);
% 6. Сравнение выходных сигналов
figure;
plot(t,yPI,'b','LineWidth',1.5);
hold on;
plot(t,yF,'r--','LineWidth',1.5);
plot(t,r,'k:','LineWidth',1.2);
grid on;
xlabel('Время, с');
ylabel('Относительная концентрация');
title('Сравнение ПИ- и нечеткого регуляторов');
legend('ПИ-регулятор', ...
```

```
    'Нечеткий регулятор', ...
    'Заданное значение');
% 7. Сравнение управляющих сигналов
figure;
plot(t,uPI,'b','LineWidth',1.5);
hold on;
plot(t,uF,'r--','LineWidth',1.5);
grid on;
xlabel('Время, с');
ylabel('Управляющий сигнал');
title('Изменение управляющих сигналов');
legend('ПИ-регулятор', ...
    'Нечеткий регулятор');
% 8. Расчет показателей качества регулирования
% Ошибки регулирования
errPI = r - yPI;
errF = r - yF;
% Интегральная абсолютная ошибка
IAE_PI = sum(abs(errPI))*Ts;
IAE_F = sum(abs(errF))*Ts;
% Перерегулирование, %
OS_PI = max(0,(max(yPI)-1)*100);
OS_F = max(0,(max(yF)-1)*100);
% Допустимое отклонение от заданного значения
tol = 0.02;
% Индексы точек за пределами допустимой полосы
idxPI = find(abs(yPI-1)>tol);
idxF = find(abs(yF-1)>tol);
```

```
% Расчет времени установления ПИ-регулятора
if isempty(idxPI)
    Ts_PI = 0;
elseif idxPI(end) < N
    Ts_PI = t(idxPI(end)+1);
else
    Ts_PI = NaN;
end
% Расчет времени установления нечеткого регулятора
if isempty(idxF)
    Ts_F = 0;
elseif idxF(end) < N
    Ts_F = t(idxF(end)+1);
else
    Ts_F = NaN;
end
% Вывод результатов для ПИ-регулятора
fprintf('\nРезультаты для ПИ-регулятора:\n');
fprintf('IAE = %.4f\n',IAE_PI);
fprintf('Перерегулирование = %.2f %%\n',OS_PI);
fprintf('Время установления = %.2f c\n',Ts_PI);
fprintf('Конечная ошибка = %.5f\n',abs(1-yPI(end)));
% Вывод результатов для нечеткого регулятора
fprintf('\nРезультаты для нечеткого регулятора:\n');
fprintf('IAE = %.4f\n',IAE_F);
fprintf('Перерегулирование = %.2f %%\n',OS_F);
fprintf('Время установления = %.2f c\n',Ts_F);
fprintf('Конечная ошибка = %.5f\n',abs(1-yF(end)));
```

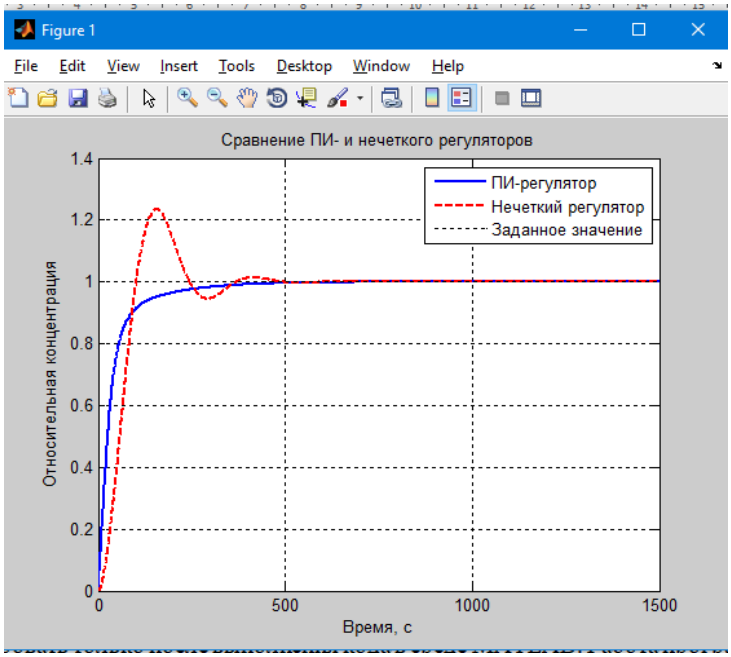


Рис.1. График сранение ПИ- и нечеткого регулятора

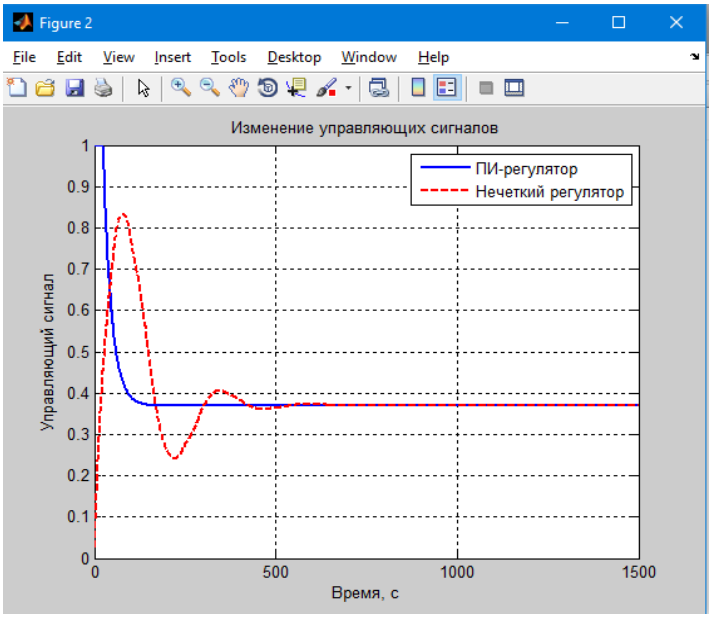


Рис.2.Изменение управляющих сигналов

Порядок использования программы: вставьте код в окно MATLAB Editor, сохраните файл, например, под именем dyeing_compare.m, и запустите его. В результате будут построены два графика: изменения концентрации и управляющих сигналов. В командном окне MATLAB отобразятся рассчитанные показатели для обоих алгоритмов управления.

Параметры Ko, To, Kp, Ki, Ku и коэффициенты нормирования, используемые в данном коде, являются исходными параметрами, принятыми для расчётов. Результаты следует фиксировать только после выполнения кода в среде MATLAB. Работа программы зависит от версии MATLAB и установленной версии Fuzzy Logic Toolbox; в зависимости от версии может потребоваться адаптация отдельных команд.

Построение систем управления в среде Simulink

Для проверки расчётных алгоритмов MATLAB в графической блочной среде используются модели Simulink. При сравнительном моделировании ПИ-регулятора и регулятора на основе нечёткой логики модель объекта, задающий сигнал, обратная связь и блоки наблюдения должны быть одинаковыми для обеих систем.

Модель с ПИ-регулятором в программе Matlab. Структура системы управления с ПИ-регулятором имеет следующий вид:

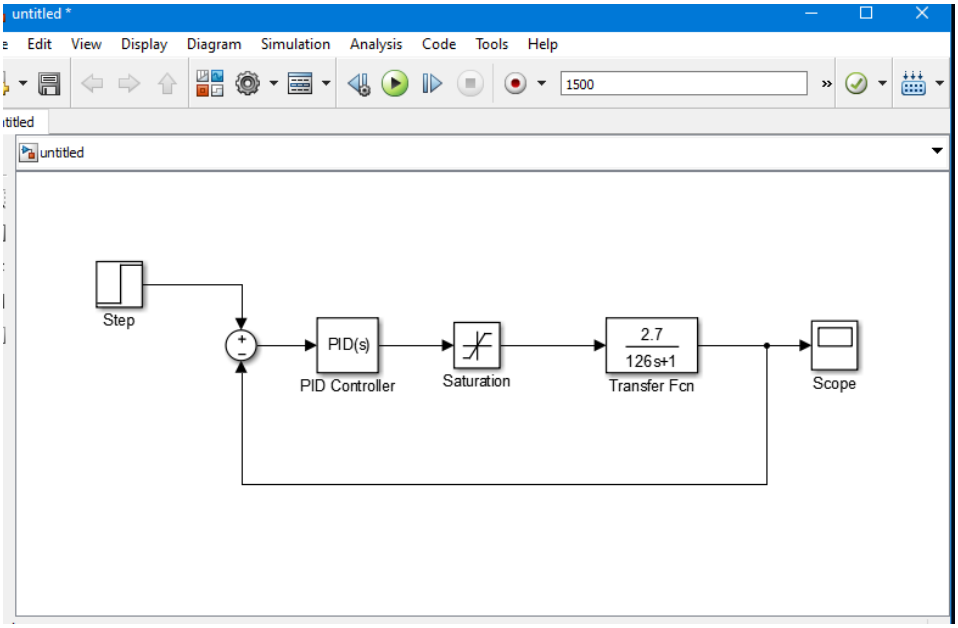


Рис. 3. Структура системы управления с ПИ-регулятором

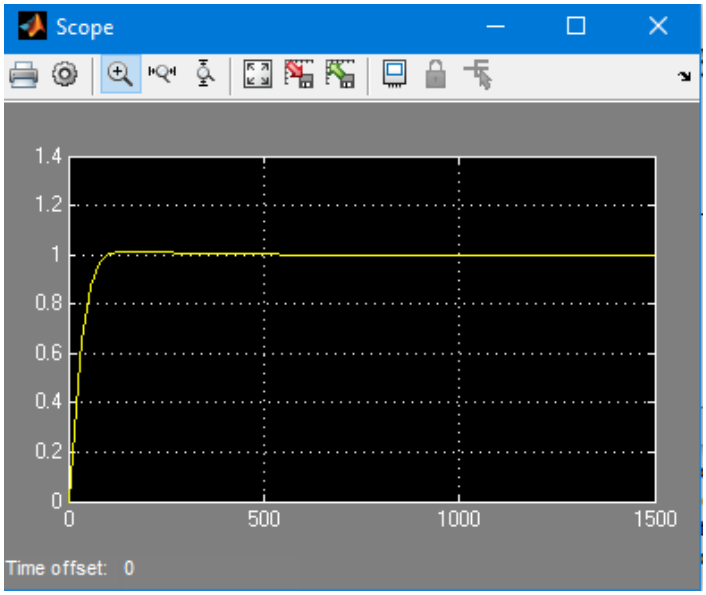


Рис.4. График системы управления с ПИ-регулятором

Блоки Simulink настроены со следующими параметрами:

Blok	**Parametr**
Step	Initial value = 0; Final value = 1
Sum	+-
PID Controller	Controller = PI
PI Controller	P = 1.94; I = 0.014
Saturation	Lower limit = 0; Upper limit = 1
Transfer Fcn	Numerator = [2.7]
Transfer Fcn	Denominator = [126 1]
Stop Time	1500

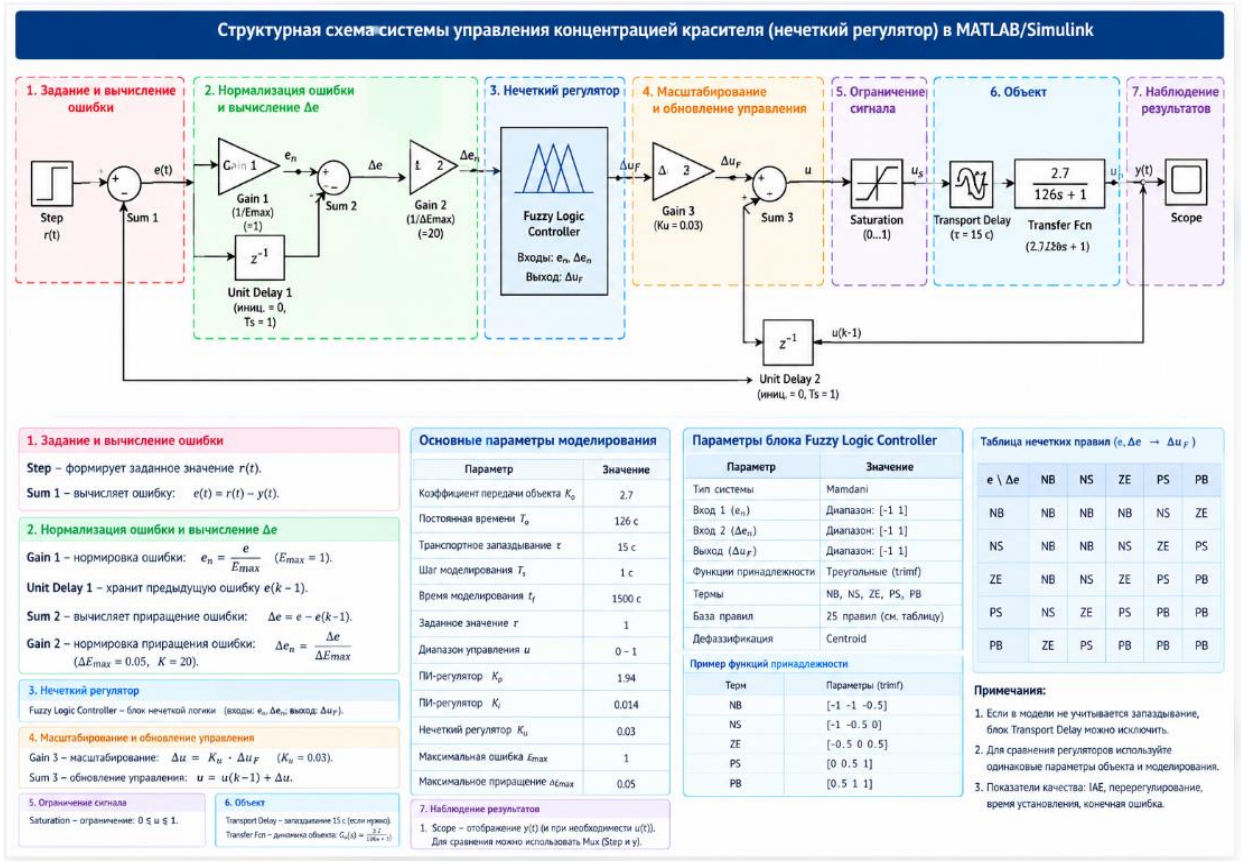
Выход блока Sum подключается ко входу ПИ-регулятора. Выход ПИ-регулятора проходит через блок Saturation, а затем поступает на вход блока Transfer Fcn. Выход объекта подключается к блоку Scope и через цепь обратной связи — к отрицательному входу блока Sum.

Важный момент: блок Saturation ограничивает управляющий сигнал, но сам по себе не устраняет интегральное насыщение (integral windup). Поэтому целесообразно реализовать алгоритм ПИ-регулятора с механизмом

ограничения интегральной составляющей, аналогичным тому, который используется в MATLAB-коде, либо применить блок ПИ-регулятора с соответствующей настройкой защиты от интегрального насыщения (anti-windup).

Модель с регулятором на основе нечёткой логики

В нечёткой системе используются два входа — ошибка регулирования и изменение ошибки. Эти сигналы обрабатываются регулятором на основе нечёткой логики, который формирует приращение управляющего сигнала.



Оценка результатов сравнительного моделирования. При сравнении ПИ-регулятора и регулятора на основе нечёткой логики недостаточно анализировать только график выходного сигнала. Для обеих систем необходимо рассчитывать одинаковые критерии качества регулирования.

Основные критерии оценки: Время регулирования. Время, необходимое для того, чтобы выходной сигнал вошёл в диапазон точности ±2% относительно заданного значения и в дальнейшем оставался в пределах этого диапазона.

Максимальное перерегулирование. Степень превышения выходным сигналом заданного значения:

$$\sigma = \frac{y_{\max} - r}{r} * 100\%.$$

Данная формула применяется для случая, когда r>0.

3. Ошибка установившегося режима.

$$e_{ss} = |r - y_{ss}|.$$

Критерий интеграла абсолютной ошибки (IAE).

$$IAE = \int_0^T |r(t) - y(t)| dt.$$

При дискретных вычислениях:

$$IAE \approx T_s \sum_{k=1}^{N} |r_k - y_k|.$$

Небольшое значение IAE указывает на малую суммарную абсолютную ошибку регулирования, накопленную за заданный промежуток времени. Однако этот показатель сам по себе не отражает всех характеристик качества системы.

Управляющий сигнал. Проверяется, не выходит ли управляющий сигнал насоса или клапана за установленные пределы и отсутствуют ли чрезмерно резкие изменения управляющего воздействия.

Ожидаемые характеристики и их проверка. Работа ПИ-регулятора зависит от двух параметров. При правильном выборе настроек он позволяет отслеживать заданное значение и уменьшать установившуюся ошибку. Однако чрезмерно высокие коэффициенты усиления могут приводить к увеличению колебаний и перерегулирования.

Регулятор Fuzzy анализирует ошибку и её изменение с помощью лингвистических правил. Его эффективность зависит от функций принадлежности, правил управления, нормирования входных сигналов и масштаба выходного сигнала.

В ходе сравнительного эксперимента можно проверить следующие гипотезы:

- Уменьшает ли нечёткое управление резкие изменения управляющего сигнала?
- Позволяет ли ПИ-регулятор достичь меньшей установившейся ошибки?
- Как изменяется качество регулирования обеих систем при изменении постоянной времени объекта?
- Ухудшаются ли показатели качества одной из систем сильнее при введении транспортного запаздывания?

Для ответа на эти вопросы рекомендуется провести дополнительные симуляции, изменяя параметры объекта и вводя внешние возмущения.

Анализ результатов. Оба рассмотренных метода управления имеют алгоритмические различия. ПИ-регулятор основан на определённом математическом законе, поэтому его настройка и анализ работы относительно просты. Нечёткое управление использует экспертные правила и формирует управляющее воздействие в соответствии с величиной ошибки и тенденцией её изменения.

Однако применение нечёткой логики не означает, что такой регулятор всегда превосходит ПИ-регулятор. Неправильно выбранные лингвистические правила или несоответствующие объекту коэффициенты нормирования могут ухудшить качество управления. Аналогично, неверно настроенный ПИ-регулятор также может давать неудовлетворительные результаты.

Ограничением настоящего исследования является представление красильной ванны моделью линейного объекта первого порядка. В реальном

процессе дополнительное влияние могут оказывать поглощение красителя волокнами, перемешивание раствора, изменение температуры и pH, а также погрешности измерительного оборудования.

Поэтому в дальнейших исследованиях целесообразно идентифицировать объект на основе экспериментальных данных, проверить модель с запаздыванием, учесть динамику дозирующего устройства и оптимизировать параметры регуляторов. Эти этапы позволят повысить практическую достоверность сравнительных результатов.

Заключение. В статье рассмотрена задача автоматического управления концентрацией красителя в процессе крашения текстильных материалов. Сформирована математическая модель объекта управления первого порядка и разработана методика сравнительного исследования ПИ-регулятора и регулятора на основе нечёткой логики.

Описаны пропорциональная и интегральная составляющие ПИ-регулятора, входные переменные, лингвистические множества и правила управления Fuzzy-регулятора. Представлен программный код для расчёта выходных сигналов обоих алгоритмов в MATLAB, сравнения графиков и оценки качества регулирования. Также изложен порядок построения отдельных моделей ПИ- и Fuzzy-управления для одного объекта в Simulink.

Для сравнительной оценки предложено использовать время регулирования, максимальное перерегулирование, установившуюся ошибку и интеграл абсолютной ошибки. Эти критерии позволяют анализировать качество систем управления в одинаковых условиях.

Основным практическим результатом работы является методика моделирования и программа MATLAB, предназначенные для проверки ПИ- и Fuzzy-регуляторов на основе одной и той же математической модели объекта. Однако для окончательного вывода о сравнительной эффективности

регуляторов необходимо завершить исследование, представив результаты вычислений, графики и заполненную таблицу показателей качества.

В дальнейшем определение параметров модели на основе данных реальной красильной ванны, учёт технологического запаздывания и экспериментальная проверка алгоритмов управления позволят повысить научную и практическую значимость исследования.

Список использованной литературы

1. Huang, C.-C., Yu, W.-H. Control of Dye Concentration, pH, and Temperature in Dyeing Processes. Textile Research Journal, 1999, 69(12), 914–918. https://doi.org/10.1177/004051759906901205.

2. Modelling and Simulation of Textile Dyeing. In: Modelling and Predicting Textile Behaviour. Woodhead Publishing, 2010, pp. 322–359. https://doi.org/10.1533/9781845697211.2.322.

3. MathWorks. Fuzzy Logic Toolbox Documentation. Fuzzy inference tizimlarini yaratish va MATLAB/Simulink muhitida modellashtirish bo‘yicha rasmiy hujjatlar. https://www.mathworks.com/help/fuzzy/.

4. MathWorks. Fuzzy Logic Controller — Simulink. Fuzzy Logic Controller blokining ishlash prinsipi va Simulink bilan integratsiyasi. https://www.mathworks.com/help/fuzzy/fuzzylogiccontroller.html.

5. MathWorks. PID Controller Tuning in Simulink. PI/PID rostlagichlarni sozlash va o‘tish jarayonini tahlil qilish bo‘yicha rasmiy hujjatlar. https://www.mathworks.com/help/control/ug/pid-controller-tuning-in-simulink.html