

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
имени академика С.П. КОРОЛЁВА»

А.В. КУЗНЕЦОВ, Г.М. МАКАРЬЯНЦ

**Технология идентификации ГТД как объекта регулирования (ОР)
в диапазоне изменения возмущающих воздействий**

Самара 2017

УДК: 681.5.015

Авторы: А.В. Кузнецов, Г.М. Макарьянц

Технология идентификации ГТД как объекта регулирования (ОР) в диапазоне изменения возмущающих воздействий: учеб. пособие / [А.В. Кузнецов, Г.М. Макарьянц]. – Самара: Изд-во Самар. ун-та, 2017. – 21 с.: ил.

В пособии описаны различные способы идентификации газотурбинного двигателя по частоте вращения по результатам натурных испытаний. Приведено описание метода проведения испытаний, обработка результатов, а так же методы получения линейных и нелинейных моделей двигателя. Дано подробное описание каждого из методов идентификации. Пособие предназначено для студентов технических специальностей и направлений.

УДК: 681.5.015

© Самарский национальный
исследовательский университет, 2017

Содержание

1. Проведение испытаний и обработка данных	4
2. Получение модели объекта управления	6
2.1. Получение линейной модели	7
2.2. Получение нелинейной модели по ускорениям ротора	9
2.3. Получение модели на основе нейронной сети	13
ПРИЛОЖЕНИЕ 1	16
ПРИЛОЖЕНИЕ 2	18
ПРИЛОЖЕНИЕ 3	20
Список использованных источников	21

1. Проведение испытаний и обработка данных

Проведение испытаний

Для построения обобщённой характеристики двигателя нужно провести натурные испытания двигателя. В качестве объекта идентификации выступает двигатель JetCat P60. В процессе испытаний необходимо получить дроссельную, разгонную и тормозную характеристики. Для определения дроссельной характеристики будем задавать определённый ступенчатый управляющий сигнал на ручке управления тягой (РУТ), величина которого была получена путём разбиения диапазона рабочих режимов на 12 частей (такое число было выбрано из удобства пользования шкалой, установленной на РУТ в этом двигателе). По величине управляющего сигнала с РУТ контроллер двигателя подаёт необходимое напряжение на насос-дозатор топлива. Во время работы двигателя информация о частоте вращения ротора и напряжении на топливном насосе записываются на встроенный в систему управления двигателем контроллер. Результаты испытаний двигателя для получения дроссельной характеристики приведены на рисунке 1.

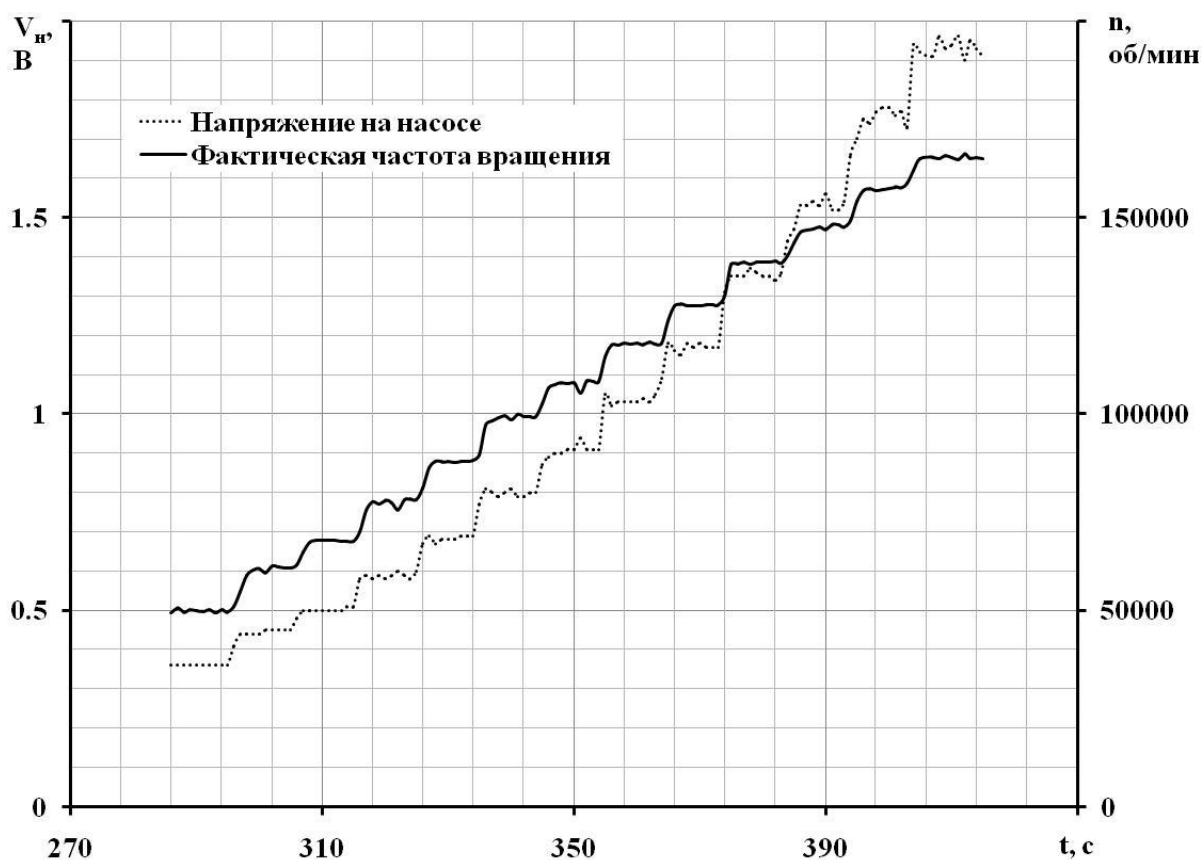


Рисунок 1 – Переходная характеристика двигателя для определения дроссельной характеристики

Разгонная и тормозная характеристики определяются путём ступенчатого перемещения РУТ из положения малого газа на максимальный режим и обратно, с небольшой паузой между этими переходными процессами (рисунок 2).

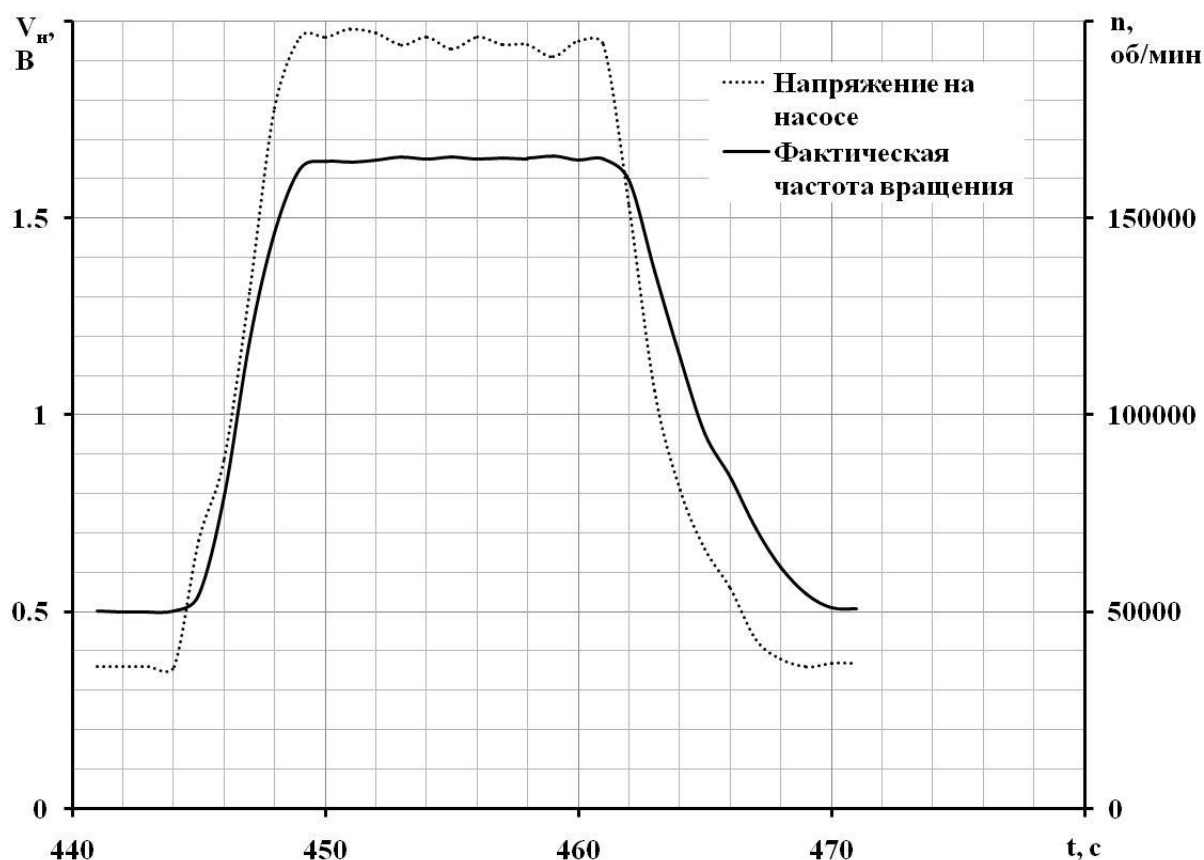


Рисунок 2 – Переходная характеристика двигателя для определения динамических характеристик двигателя

Обработка данных

Для дальнейшей работы над моделью двигателя полученные результаты необходимо свести в график обобщённой характеристики двигателя, показывающий зависимость расхода топлива от частоты вращения, а так же ускорения ротора от частоты вращения. Расход топлива определяется по напряжению, исходя из условия, что на 1 В напряжения, подаваемого на насос, расход топлива равен 1,633 г/с. Данные о расходе были получены из информации о технических характеристиках двигателя.

Для построения дроссельной характеристики берутся значения расхода топлива и частоты вращения на установившихся режимах. Для разгонной и тормозной характеристик значения берутся с графика переходных процессов. Для определения ускорения в некоторой точке значение частоты вращения ротора двигателя в этой точке вычитается из значения в следующей за ней точке:

$$dn/dt_i = n_{i+1} - n_i$$

В крайних точках графика ускорений примем её значение равным нулю (рисунок 3).

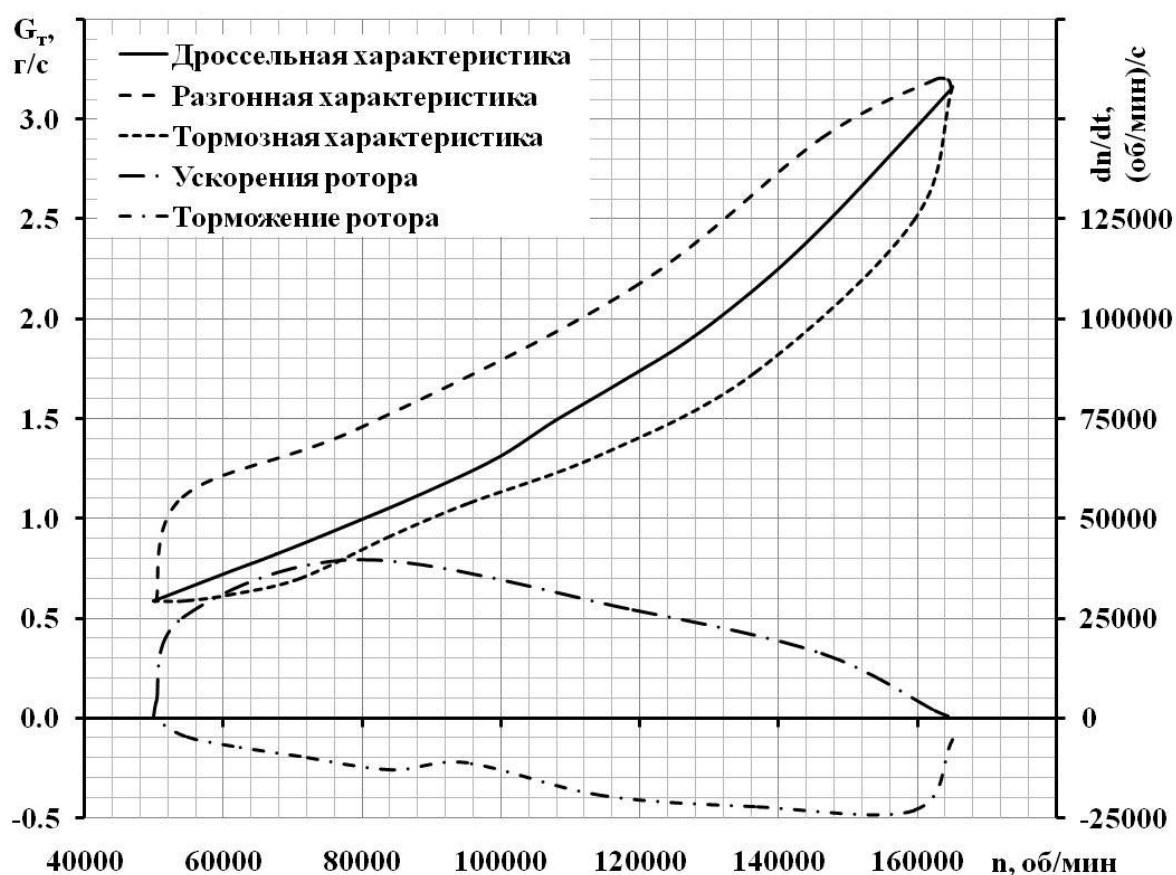


Рисунок 3 – Полученная обобщённая характеристика двигателя

Полученная характеристика необходима для получения математической модели двигателя.

2. Получение модели объекта управления

В процессе получения математической модели объекта управления по частоте вращения можно использовать разные способы, однако в рамках данной работы рассмотрим три из них. Первый способ позволяет получить линейные модели двигателя на каких-либо установившихся режимах. Второй способ позволяет получить многорежимную динамическую модель. Третий способ представляет собой использование нейронных сетей. Эти модели являются упрощёнными и не отражают в себе структуры двигателя, однако из-за своей простоты они не потребляют много вычислительных ресурсов. У

каждой из этих моделей есть свои плюсы и минусы, которые определяют их область применения.

Модель, полученная первым способом, представляет собой передаточную функцию объекта управления на каком-то установившемся режиме. Эту модель можно использовать при создании регулятора поддержки какого-либо режима работы и с ней можно работать аналитически, однако эта модель не может полностью отражать поведение двигателя в широком диапазоне переходных режимов. Выходом могло бы служить использование нескольких передаточных функций, либо передаточная функция с возможностью динамического изменения своих коэффициентов и постоянных времени, однако в процессе решения этой задачи этот способ не оправдал себя по точности полученных результатов.

Модель, полученная вторым способом, позволяет получить достаточно точное представление о динамике объекта, так как является нелинейной, при этом по скорости расчёта эта модель не уступает линеаризованной. Однако, аналитически исследовать эту модель будет затруднительно.

В последнем способе нейросетевая модель может использоваться для получения недостающей информации для системы управления (например, для оценки расхода воздуха, который зависит от многих параметров или для определения тяги двигателя), но большинство статей в российских и зарубежных журналах по этой теме показывают различные способы определения неисправности двигателя или его отдельных систем.

Чтобы выбрать один из способов создания математической модели необходимо знать для чего эта модель будет использоваться. Если нужно оценить устойчивость двигателя на каком-то из режимов, то лучше будет воспользоваться первым способом, а если нужно оценить работу системы управления в широком диапазоне частот вращения ротора, то лучше воспользоваться вторым или третьим способом.

2.1. Получение линейной модели

Подробный вывод уравнений и описание систем управления приведён в [1]. Стоит отметить, что в приведённом учебном пособии предлагается использовать параметры в отклонениях, однако в данной работе нам нужны полные параметры и поэтому приведённые здесь уравнения будут записаны в полных параметрах.

Наиболее полная картина динамики двигателя получается при использовании разгонных характеристик, поэтому изначально эксперимент был проведён для этого метода. Сущность метода заключается в том, что в процессе

приёмности регистрируются значения частоты вращения ротора двигателя n и расход топлива G_m . Затем по частоте вращения рассчитывается ускорения ротора dn/dt . Полученные данные заносятся на один график со статической характеристикой (рисунок 4).

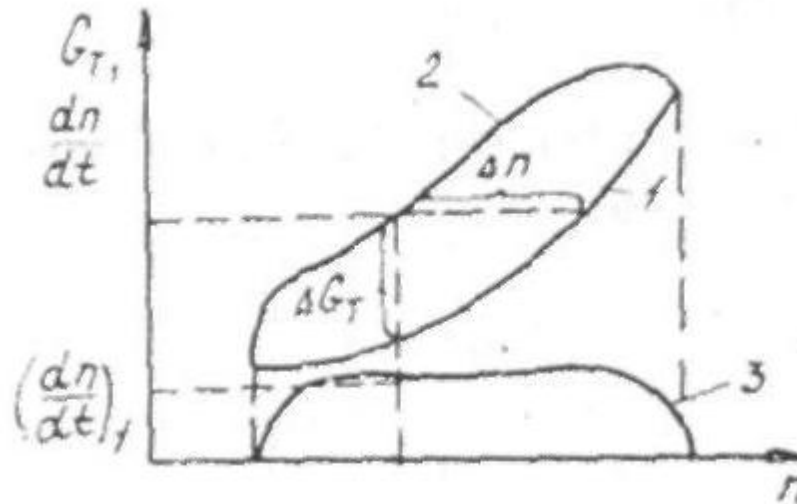


Рисунок 4 – Разгонная характеристика ТРД:

1. Статическая (дрессельная) характеристика;
2. Подача топлива при приёмности двигателя;
3. Ускорение ротора двигателя.

Если провести горизонтальную прямую, пересекающую дроссельную характеристику и кривую разгона 2 (рисунок 4), и рассматривать неустановившийся режим как отклонение от установившегося при $G_m = const$, то можно определить постоянную времени:

$$T_d = -\Delta n / \left(\frac{dn}{dt} \right)$$

Коэффициент усиления двигателя определяется по его дроссельной характеристике как котангенс угла наклона кривой:

$$K = \left(\frac{dn}{dG_T} \right) = \cot \Psi$$

где Ψ – угол наклона графика дроссельной характеристики к горизонтальной оси в исследуемой точке. Котангенс – отношение прилежащего катета прямоугольного треугольника (разность частот вращения) к противолежащему

(разность расходов топлива или напряжения на топливном насосе). Гипотезой в данном случае будет дроссельная характеристика, поэтому нужно стараться выбрать достаточно маленький отрезок кривой, чтобы он был наиболее приближен к прямой, однако не следует брать его слишком маленьким – трудно будет измерить данные с графика.

Полученные по приведённым формулам расчёты, основанные на обобщённой характеристике (рисунки 3, 4), представлены в таблице 1. За базовые значения приняты значения соответствующих параметров на дроссельной характеристике. Для этого метода нам нужна только дроссельная и разгонная характеристики.

Таблица 1 – Результаты расчёта

$G_{т.баз}$	$G_{т.р}$	$n_{баз}$	n_p	Δn	$\cot \Psi$	(dn/dt)	K_{nG}	T_d
0.6666	1.1105	54170	89139	34969	115282	25230	115282	1.3860
0.9562	1.4534	79400	106937	27537	66145	39660	66145	0.6943
1.7322	2.1556	119060	135040	15980	39607	27190	39607	0.5877
2.4906	2.9068	146250	158916	12666	31063	16170	31063	0.7833
3.0290	3.2007	162420	167203	4783	27531	1930	27531	2.4781

2.2. Получение нелинейной модели по ускорениям ротора

Получение нелинейной модели в некотором смысле проще, потому что нет никаких ручных вычислений, а все данные берутся из графика обобщённой характеристики (рисунок 3). Для кривых разгона и торможения необходимы соответствующие графики ускорения и торможения ротора (ускорение ротора по дроссельной характеристике равно нулю). В прошлом разделе описывалось получение значения ускорения для разгона ротора – аналогично рассчитываются значения для торможения.

Модель строится на основе экспериментальной зависимости ускорения ротора от частоты вращения и расхода топлива на основе обобщённой характеристики двигателя. Для этого обобщённая характеристика рассекается горизонтальными линиями, соответствующими постоянным величинам расхода топлива. Эти линии постоянных расходов пересекают три кривых – кривую разгона, дроссельную кривую и кривую торможения. Из точек пересечения берутся значения частоты вращения ротора и соответствующее для этой частоты ускорение. Все полученные данные сведены в таблицу 2.

Таблица 2 – Зависимость ускорения ротора от частоты вращения и расхода топлива

	$G_{т.пр}$, г/с	Ускорение ротора	Дроссельная характеристика	Торможение ротора
$n_{пр}$, об/мин	0,6	49907	49907	49907
$dn_{пр}/dt$, (об/мин)/с		0	0	0
$n_{пр}$, об/мин	1,0	52000	80000	90000
$dn_{пр}/dt$, (об/мин)/с		20000	0	-11000
$n_{пр}$, об/мин	1,5	82000	108000	126000
$dn_{пр}/dt$, (об/мин)/с		40000	0	-21000
$n_{пр}$, об/мин	2,0	112000	132000	146000
$dn_{пр}/dt$, (об/мин)/с		30000	0	-24000
$n_{пр}$, об/мин	2,5	132000	148000	160000
$dn_{пр}/dt$, (об/мин)/с		22500	0	-22500
$n_{пр}$, об/мин	3,0	150000	160000	164000
$dn_{пр}/dt$, (об/мин)/с		13000	0	-10000
$n_{пр}$, об/мин	3,2	164895	164895	164895
$dn_{пр}/dt$, (об/мин)/с		0	0	0

Приведённые в таблице 2 данные линейно интерполируются. Визуализация линейной интерполяции представлена на рисунке 5, где пунктирной линией обозначена область работы двигателя.

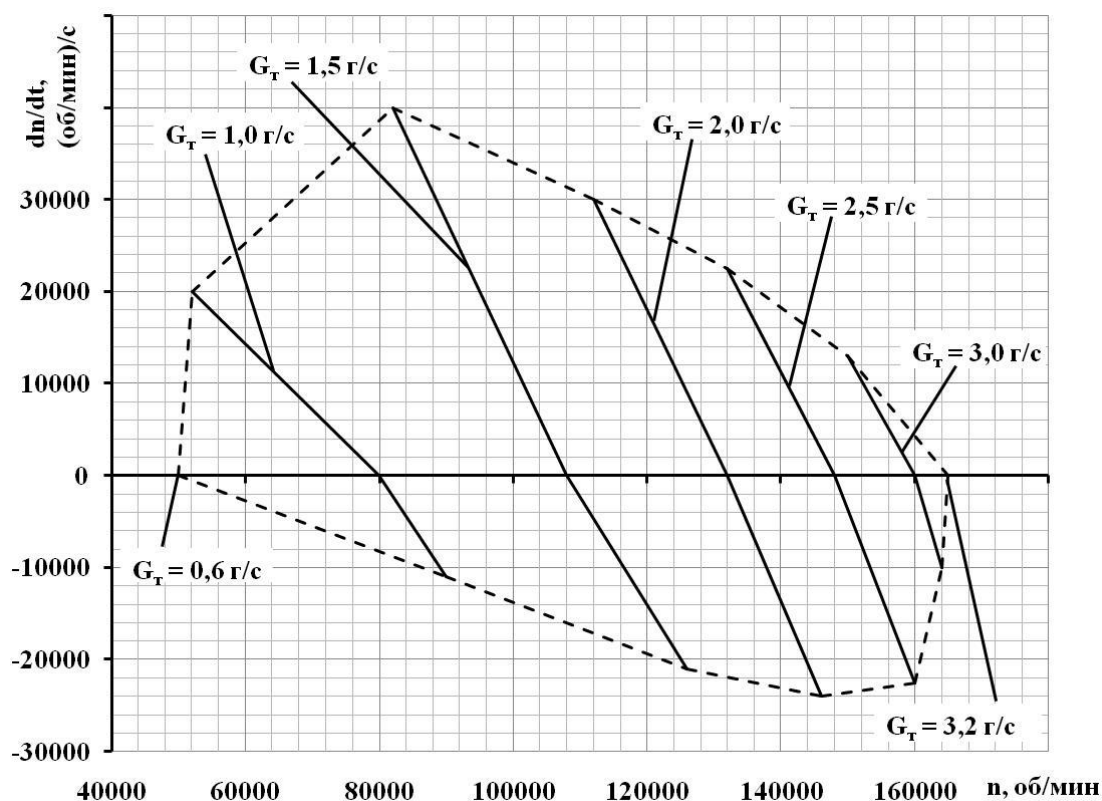


Рисунок 5 – Ускорения ротора в зависимости от частоты вращения и расхода топлива

Функция для линейной интерполяции двумерных массивов реализована в MatLab/Simulink в виде блока Matlab Function (ПРИЛОЖЕНИЕ 1). По введённым в этот блок значениям расхода топлива и частоты вращения рассчитывается соответствующее ускорение ротора двигателя, которое затем интегрируется и в виде обратной связи поступает в интерполяционную функцию в виде частоты вращения (рисунок 6). Начальное значение интегратора равно частоте вращения на малом газе и составляет 50000 об/мин.



Рисунок 6 – Нелинейная модель двигателя в MatLab/Simulink

Для сравнения модели с реальным двигателем была собрана система, представленная на рисунке 7.

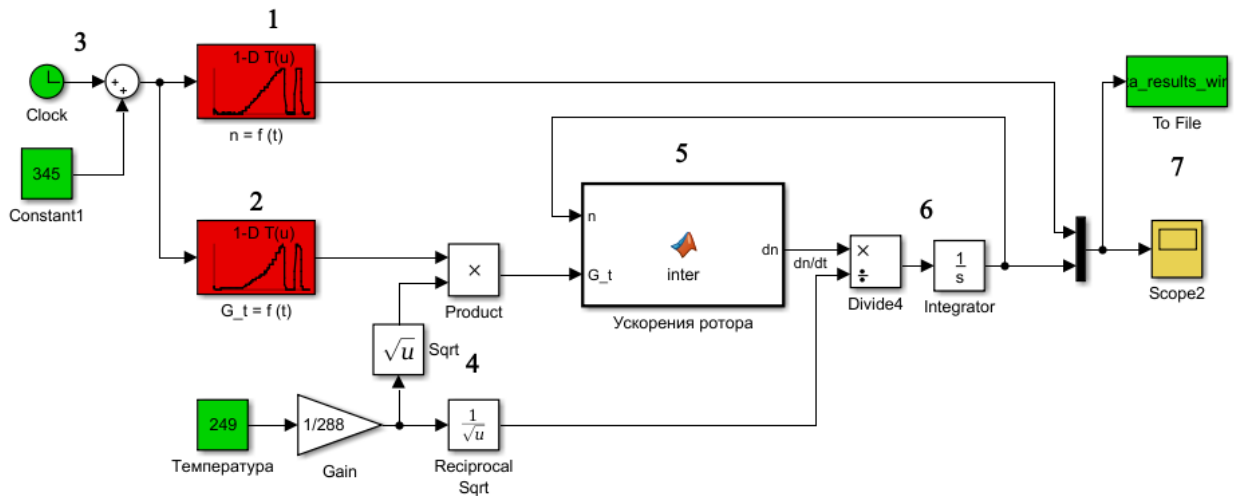


Рисунок 7 – Проверка работоспособности модели.

Модель двигателя в MatLab/Simulink (рисунок 7) работает следующим образом: в табличных блоках 1 и 2 занесены данные по частоте вращения и расходу топлива, полученные в результате испытаний в зависимости от времени 3; коэффициент перевода физических величин в приведённые реализован набором блоков 4; интерполяция производится в блоке 5, результат вычислений которого переводится из приведённых значений в физические в блоках 6; данные из блоков 1 и 6 передаются в 7 для записи в файл и визуализации. В создании модели использовались данные, полученные с 275-й секунды и до 480-й, поэтому начало отсчёта задаём блоком Constant1, а время

расчёта устанавливаем на 205 секунд. Расход топлива и частота вращения представлены как блоки 1-D Lookup Table, в которых приведена зависимость расхода топлива и частоты вращения от времени. Для записи в файл добавляется блок To File из стандартной библиотеки. В нём прописывается название файла Data_results.mat и название переменной Data_results. Сохранять в формате Timeseries. Для автоматической обработки данные используем следующий скрипт:

```
load Data_results

time = Data_results(:,1);
real = Data_results(:,2);
model = Data_results(:,3);

figure; plot(time, real, time, model);

error = (real - model) ./ real .* 100;

figure; plot(time, error);

mse = sqrt (sum ((model - real) .^ 2) / length (real));
me = sum(abs(error)) / length(error);
```

Этот скрипт автоматически создаст графики (рисунки 8, 9).

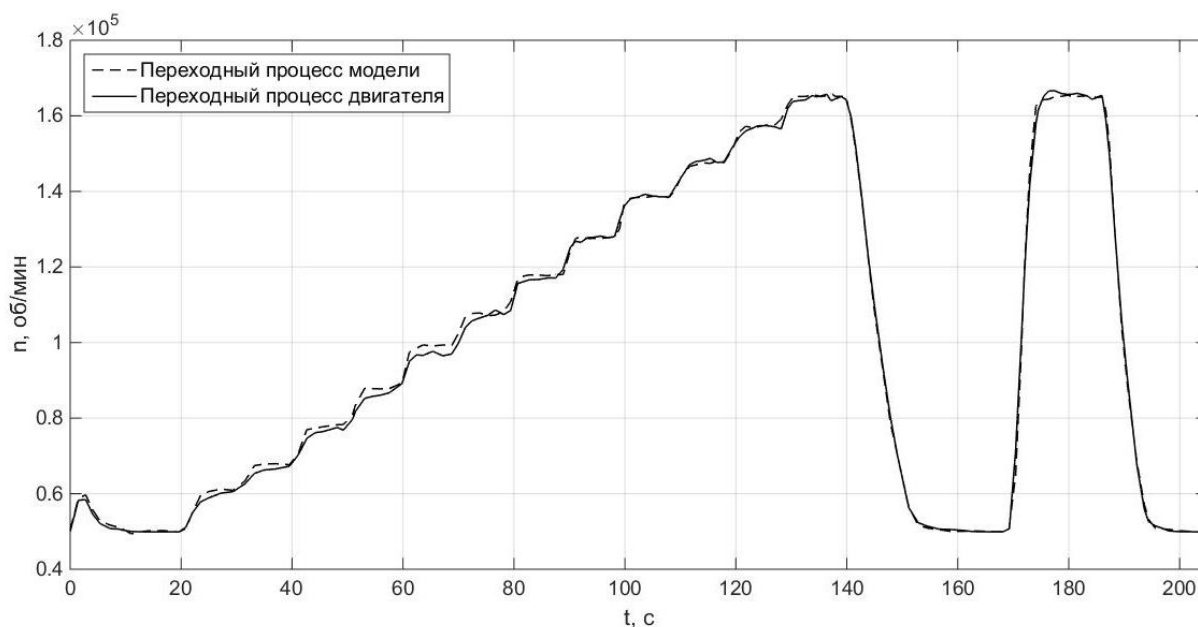


Рисунок 8 – Сравнение переходных и установившихся процессов модели и двигателя

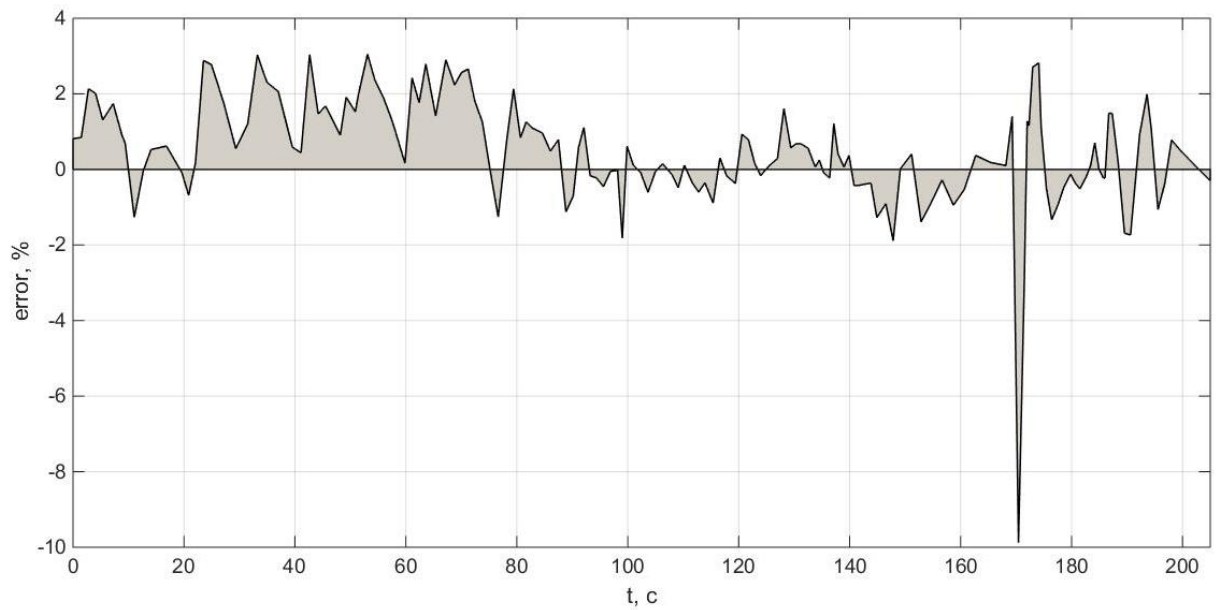


Рисунок 9 – Величина рассогласования модели по времени

Средняя ошибка полученной модели по времени моделирования (me) составила 1,0381 %, а среднеквадратичное отклонение (mse) – 1356,1 об/мин.

2.3. Получение модели на основе нейронной сети

Полученные в результате испытаний двигателя данные собраны с достаточно большим интервалом (1 секунда). Для обучения нейронной сети с требуемым шагом по времени линейно интерполируем полученные в результате эксперимента данные с помощью 1-D Lookup Table, как в предыдущем разделе (рисунок 10).

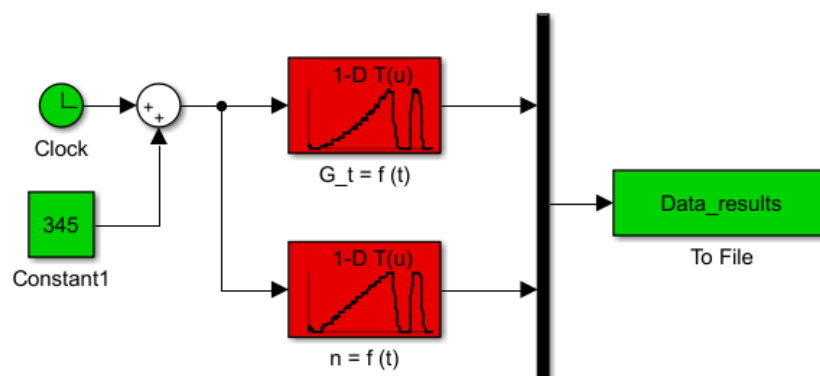


Рисунок 10 – Получение обучающей выборки

В качестве обучающей выборки был взят набор данных, представленный на рисунке 2, а в качестве тестовой выборке был взят весь набор значений, полученный в результате испытаний. Загрузка данных из полученного файла и создание динамической нейронной сети представлены в ПРИЛОЖЕНИИ 2 с наличием подробных комментариев. В Matlab следует создать новый скрипт, скопировать в него текст из ПРИЛОЖЕНИЯ 2 и сохранить с названием `Script_NN.m`. Для получения наиболее точных результатов следует провести несколько запусков с разным количеством нейронов в скрытом слое. Это связано с тем, что весовым коэффициентам нейронов в начальный момент времени присваивается случайное значение в определённом диапазоне и при создании нейронных сетей с одинаковой структурой могут быть некоторые отличия в точности идентификации. Скрипт, отвечающий за проведение необходимого числа запусков с требуемым диапазоном количества скрытых нейронов приведён в ПРИЛОЖЕНИИ 3. Его следует так же сохранить в виде отдельного скриптового файла с каким-либо удобным названием. В результате работы скрипта из ПРИЛОЖЕНИЯ 3 создаются графики, приведённые на рисунке 11, а так же набор файлов с названием `net` и различными индексами, в которые сохраняются полученные в результате идентификации нейронные сети.

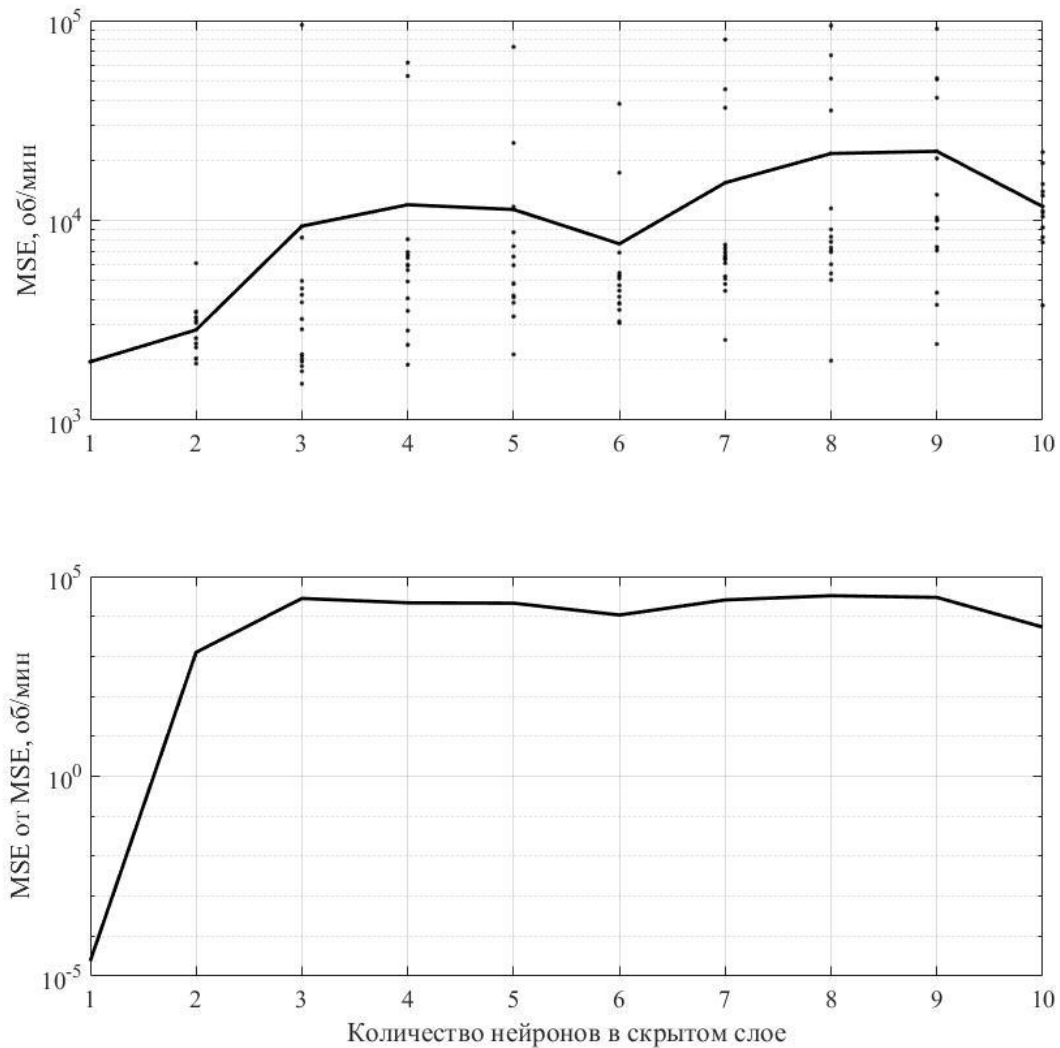


Рисунок 11 – Распределение среднеквадратичных отклонений

Из рисунка 11 можно определить какую нейронную сеть следует выбрать для дальнейшей работы. В данном случае это любая сеть в одном скрытом нейроне, так как она обладает наименьшей величиной ошибки и малым разбросом этих ошибок при каждом запуске. Добавив выбранную нейронную сеть в рабочую область, можно передать ей как в функцию входной набор данных (расход топлива, взятый из тестовой выборки) и построить графики переходных процессов и величину ошибки по времени командой plot (рисунок 12).

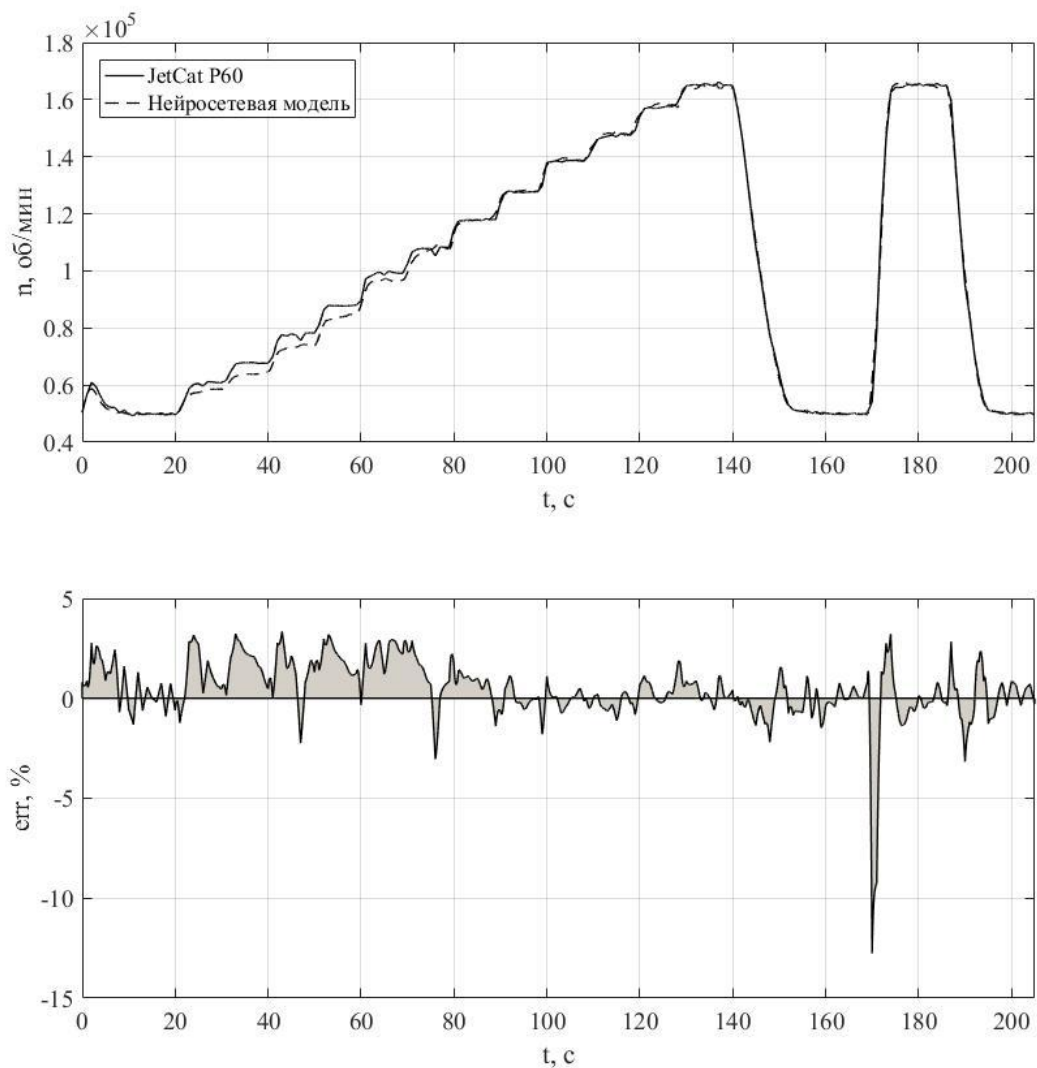


Рисунок 12 – Валидация нейросетевой модели

Средняя ошибка полученной модели по времени моделирования составила 1,0317 %, а среднеквадратичное отклонение– 1960,5 об/мин.

ПРИЛОЖЕНИЕ 1

```
function dn = inter(G_t,n)

mas = [0.5879    1        1.5      2        2.5      3        3.154    ...
       49907    52000    82000    112000   132000   150000   164895   ...
       49907    80000    108000   132000   148000   160000   164895   ...
       49907    90000    126000   146000   160000   164000   164895   ...
       0        20000    40000    30000    22500    13000    0        ...
       0        0        0        0        0        0        0        ...
       0        -11000   -21000   -24000   -22500   -10000   0        ];

M = 6;
N = 7;
N1 = M/2;
Gmas = zeros(1,M);
k1 = 0;
k2 = 0;
XA = 0;
XB = 0;
dn = 0;

if G_t>mas(N)
    kp=1+N-2;
    ks=1+N-1;
    XA=mas(kp);
    XB=mas(ks);
    k1=1-2+N*2;
    k2=1-1+N*2;
    for i=1:1:M
        kq=k1+(i-1)*N;
        kw=k2+(i-1)*N;
        if XB==XA
            XB=XB+0.1;
        end
        Gmas(i)=mas(kq)+((mas(kw)-mas(kq))*(G_t-XA))/(XB-XA);
    end
end

if G_t<mas(1)
    XA=mas(1);
    XB=mas(2);
    k1=N+1;
    k2=N+2;
    for i=1:1:M
        kq=k1+(i-1)*N;
        kw=k2+(i-1)*N;
        if XB==XA
            XB=XB+0.1;
        end
        Gmas(i)=mas(kq)+((mas(kw)-mas(kq))*(G_t-XA))/(XB-XA);
    end
end

else
    for i=1:1:N-1
        if G_t>=mas(i)
            if G_t<=mas(i+1)
                XA=mas(i);
                XB=mas(i+1);
                k1=N+i;
                k2=1+N+i;
            else
                continue;
            end
        end
    end
end
```

```

        end
    end

    for i1=1:1:M
        kq=k1+(i1-1)*N;
        kw=k2+(i1-1)*N;
        if XB==XA
            XB=XB+0.1;
        end
        Gmas(i1)=mas(kq)+(((mas(kw)-mas(kq))*(G_t-XA))/(XB-XA));
    end
end
if n>Gmas(N1)
    kp=1+N1-2;
    ks=1+N1-1;
    XA=Gmas(kp);
    XB=Gmas(ks);
    k1=1-2+N1*2;
    k2=1-1+N1*2;
    kq=k1;
    kw=k2;
    if XB==XA
        XB=XB+0.1;
    end
    dn=Gmas(kq)+(((Gmas(kw)-Gmas(kq))*(n-XA))/(XB-XA));
end
if n<Gmas(1)
    XA=Gmas(1);
    XB=Gmas(2);
    k1=N1+1;
    k2=N1+2;
    kq=k1;
    kw=k2;
    if XB==XA
        XB=XB+0.1;
    end
    dn=Gmas(kq)+(((Gmas(kw)-Gmas(kq))*(n-XA))/(XB-XA));
else
    for i=1:1:N1-1
        if n>=Gmas(i)
            if n<=Gmas(i+1)
                XA=Gmas(i);
                XB=Gmas(i+1);
                k1=N1+i;
                k2=1+N1+i;
            else
                continue;
            end
        end
        kq=k1;
        kw=k2;
        if XB==XA
            XB=XB+0.1;
        end
        dn=Gmas(kq)+(((Gmas(kw)-Gmas(kq))*(n-XA))/(XB-XA));
    end
end
end
end

```

ПРИЛОЖЕНИЕ 2

```
%% Исходные данные

init_in = Data_results(2,16000:end); % Входные данные для обучения
init_out = Data_results(3,16000:end); % Выходные данные для обучения
test_in = Data_results(2,:); % Входные тестовые данные
test_out = Data_results(3,:); % Выходные тестовые данные

%% Создание и обучение НС

X = tonndata(init_in,true,false);
T = tonndata(init_out,true,false);

trainFcn = 'trainlm'; % Тренировочная функция
inputDelays = 1:1; % Задержка входа
feedbackDelays = 1:1; % Задержка выхода
%hiddenLayerSize = 8; % Количество скрытых нейронов

net = narxnet(inputDelays,feedbackDelays,hiddenLayerSize,'open',trainFcn);

% Подготовка обучающих данных для динамической НС:
[x,xi,ai,t] = preparets(net,X,{},T);

net.divideParam.trainRatio = 100/100; % Количество данных для тренировки
net.divideParam.valRatio = 0/100; % Количество данных для валидации
net.divideParam.testRatio = 0/100; % Количество данных для тестирования
%net.divideFcn = 'divideblock';
net.performFcn = 'mse'; % Функция отклонений
%net.performParam.regularization = 1;

[net,tro] = train(net,x,t,xi,ai); % Обучение сети
y = net(x,xi,ai);
performanceInitO = perform(net,t,y);
%view(net)

net = closeloop(net); % Преобразование сети в сеть с замкнутым контуром
[x,xi,ai,t] = preparets(net,X,{},T);
[net,trc] = train(net,x,t,xi,ai);
y = net(x,xi,ai); % Значение выхода сети на обучающей выборке
%e = gsubtract(t,y); % Отклонение от обучающей выборки
performanceInitC = perform(net,t,y); % Среднеквадратичное отклонение по обучающей выборке

%% Проверка на тестовой выборке

Xt = tonndata(test_in,true,false);
Tt = tonndata(test_out,true,false);

% Подготовка тестовых данных для динамической НС
[xt,xit,ait,tt] = preparets(net,Xt,{},Tt);

output = cell2mat(net(xt,xit,ait)); % Вектор выходных значений сети
target = cell2mat(tt); % Конвертирование данных
errors = (target - output) ./ target .* 100; % Вектор ошибок в процентах
meanErr = sum(abs(errors)) / length(errors); % Среднее значение ошибок в процентах

% Среднеквадратичное отклонение
performanceTest = sqrt(sum((target - output).^2) / length(output));
```

ПРИЛОЖЕНИЕ 3

```
%% Запуск расчёта

errList = [];
hiddens = [];
%timeList = [];
%epochList = [];

lowHid = 1;
highHid = 10;
num_starts = 15;

for n = 1:num_starts
    for i = lowHid : highHid
        hiddenLayerSize = i;
        run Script_NN;
        %timeList(n,i) = time_learn;
        %epochList(n,i) = epoch_learn;
        errList(n,i) = performanceTest;
        hiddens(i) = i;
        save(['net_' num2str(n) '_' num2str(i)], 'net');
    end
end

%% Построение графиков
for k = lowHid:highHid
    meanErrList(k) = sum(errList(:,k))/num_starts;
    mseErrList(k) = sqrt(sum((errList(:,k) - meanErrList(k)).^2) / highHid);
end

% Настройка шрифтов в подписях к осям
set(0, 'DefaultAxesFontSize', 14, 'DefaultAxesFontName', 'Times New Roman');
% Настройка шрифтов к тексту на графиках
set(0, 'DefaultTextFontSize', 14, 'DefaultTextFontName', 'Times New Roman');
% Настройка размера графика и его положение на экране
figure('Units', 'normalized', 'OuterPosition', [0 0 1 1]);

figure;
subplot(2,1,1), semilogy(hiddens, errList, 'k.', hiddens, meanErrList, 'k.-', 'LineWidth', 2);
grid on;
title('График среднего среднеквадратичного отклонения');
xlabel('number of hidden neurons');
ylabel('MSE of model, rps');

subplot(2,1,2), semilogy(hiddens, mseErrList, 'k', 'LineWidth', 2);
grid on;
title('MSE от MSE модели');
xlabel('number of hidden neurons');
ylabel('MSE of MSE');

%% Очистка переменных
clear init_in init_out test_in test_out
clear X T trainFcn inputDelays feedbackDelays hiddenLayerSize
clear x xi ai t
clear y e
clear xc xic aic tc yc
clear xs xis ais ts ys
clear Xt Tt xt xit ait tt
clear output target errors
```

Список использованных источников

1. Гимадиев А.Г., Е.В. Шахматов, В.П. Шорин Системы автоматического регулирования авиационных ГТД. Куйбышев.: Куйбыш. авиац. ин-т, 1990. 122 с.
2. Кулагин В.В. Теория, расчёт и проектирование авиационных двигателей и энергетических установок. М.: Машиностроение, 2003. 616 с.
3. Добрянский Г.В., Мартьянова. Т.С. Динамика авиационных ГТД. М.: Машиностроение, 1989. 240 с.
4. А.Ю. Ткаченко, В.Н. Рыбаков, И.Н. Крупенич, Я.А. Остапюк, Е.П. Филинов Автоматизированная система для виртуальных испытаний газотурбинных двигателей // Вестник Самарского государственного аэрокосмического университета. 2014. № 5(47), ч. 3. С. 113-119.
5. Гольберг Ф.Д., Батенин А.В. Математические модели газотурбинных двигателей как объектов управления. М.: Московский авиационный институт, 1999. 82 с.
6. Шевяков А.А. Автоматика авиационных и ракетных силовых установок. М.: Машиностроение, 1965. 548 с.
7. <https://habrahabr.ru/post/195842/>