|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский авиационный институт**  **(национальный исследовательский университет)»** |

Институт № 3 **«Системы управления, информатика и электроэнергетика»**

Кафедра №301 **«Системы автоматического и интеллектуального управления»**

Направление подготовки **24.05.06 «Системы управления летательными аппаратами»**

**Отчет**

|  |
| --- |
| **по курсовому проекту** |

**Название:**

Цифровая система управления короткопериодическим боковым движением самолёта

**Дисциплина:** Теория цифровых систем управления

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | М30-401С-18 |  |  | С.С. Дзуцев |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | В.Д. Белоногов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

**Оглавление**

[**Цель работы** 3](#_Toc103921945)

[**Постановка задачи** 3](#_Toc103921946)

[**Технические требования** 4](#_Toc103921947)

[**Числовые данные** 6](#_Toc103921948)

[**1.** **Анализ динамики объекта управления** 7](#_Toc103921949)

[**2.** **Выбор параметров цифрового закона управления в контуре демпфирования** 16](#_Toc103921950)

[**3.** **Построение области устойчивости упрощённой системы в плоскости параметров , .** 30](#_Toc103921951)

[**4.** **Выбор параметров закона управления в канале элеронов** 33](#_Toc103921952)

[**5.** **Выбор коэффициента электрической связи от ручки лётчика** 37](#_Toc103921953)

[**6.** **Проверка заданных требований** 38](#_Toc103921954)

[**7.** **Полный алгоритм управления боковым движением** 41](#_Toc103921955)

[**Заключение** 45](#_Toc103921956)

[**Список иллюстраций** 46](#_Toc103921957)

[**Список использованных источников** 46](#_Toc103921958)

[**Приложение 1** 47](#_Toc103921959)

[**Приложение 2** 49](#_Toc103921960)

[**Приложение 3** 53](#_Toc103921961)

[**Приложение 4** 55](#_Toc103921962)

[**Приложение 5** 60](#_Toc103921963)

# **Цель работы**

Исследовать динамику цифровой системы управления, разработать структуру и выбрать параметры закона управления

# **Постановка задачи**

1. Провести анализ динамики объекта управления, сделать выводы о динамических свойствах объекта. Обосновать необходимость улучшения характеристик управляемости в боковом канале. Сформулировать цели управления.
2. Выбрать параметры цифрового закона управления в контуре демпфирования движения рыскания, исходя из желаемого распределения корней характеристического уравнения в упрощенной системе плоского бокового движения. Рекомендуемая последовательность действий:
   1. Составить уравнения цифровой системы в канале рыскания с учетом динамики исполнительных устройств.
   2. Получить дискретные передаточные функции и уравнение замкнутого цифрового контура.
   3. Выбрать параметры цифрового регулятора.
   4. Построить переходные процессы и убедится в выполнении требований к системе.
   5. Выбрать структуру и параметры дискретного наблюдателя, обеспечивающего измерение полного вектора состояния по доступным измерениям. Построить процессы управления по оцененным наблюдателем координатам состояния.
3. Построить область устойчивости упрощенной системы демпфирования рыскания в плоскости двух параметров ЦАУ, заданных преподавателем.
4. Выбрать параметры закона управления в канале элеронов, обеспечивающего заданные требования:
   1. Сформировать полную модель бокового движения самолета,
   2. Сформировать дискретную модель бокового движения с учетом выбранного управления в канале рыскания,
   3. Построить частотную характеристику цифрового контура демпфирования крена,
   4. Выбрать параметры ЦАУ в контуре демпфирования крена.
5. Выбрать коэффициент электрической связи от ручки летчика.
6. Осуществить проверку выполнения полного набора требований к системе при использовании синтезированных законов управления.
7. Сформировать и оформить полный алгоритм управления боковым движением и общую логику его работы.

# **Технические требования**

1. Затухание короткопериодических колебаний при управлении по и должно быть не менее, чем в 10 раз за период колебаний.
2. Собственная частота колебаний по и должна быть не менее .
3. Переходный процесс при управлении от ручки летчика должен иметь практически монотонный характер, допустимое перерегулирование 5%. Время процесса не должно превышать ,
4. Расход ручки на приращение в установившемся режиме управления по должен соответствовать величине .
5. Параметры закона управления должны обеспечивать не менее, чем двукратный запас устойчивости по любому коэффициенту .
6. Алгоритмы цифрового управления должны обладать свойствами собственной устойчивости и реализуемости.

**Математические модели элементов цифровой системы.**

Структурная схема цифровой системы управления короткопериодическим боковым движением самолёта представлена на рисунке 1.

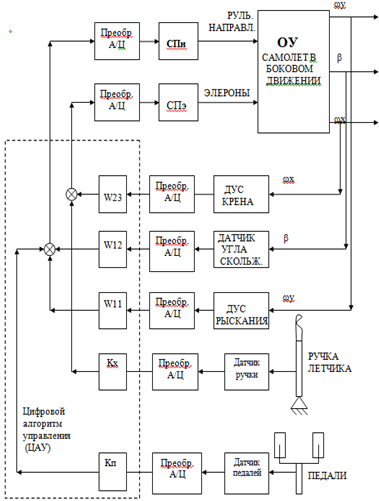
****

Рисунок 1. Структурная схема цифровой системы управления

Упрощенные уравнения короткопериодического бокового движения самолета имеют вид:

Где , , β – угловые скорости вращения самолета и угол скольжения соответственно.

Уравнения исполнительных устройств – сервопривода руля направления и элеронов имеют вид:

Где , – управляющие воздействия на сервоприводы рулей,

, – постоянные времени исполнительных устройств.

Цифровой алгоритм управления, реализуемый на БЦВМ, формируется в виде линейной комбинации координат системы, измеренных датчиками и задающих воздействий от летчика.

Где и – команды управления вырабатываемые системой управления,

и – задающие воздействия от ручки и педалей,

и - константы алгоритмы управления.

# **Числовые данные**

# **Анализ динамики объекта управления**

Для расчётов используем среду MATLABR2021b.

Имея систему уравнений динамики бокового движения можем преобразовать её к матричному виду:

Выпишем известные нам матрицы:

Тогда имеем

Где

– наблюдаемые величины.

Матрицу передаточных функций можно получить, используя соотношение 6. [1]

Где – диагональная единичная матрица размера ;

Произведём расчёт в среде MATLAB (приложение 1) и последовательно выпишем все передаточные функции системы.

Для построения переходных функций по заданным передаточным функциям при ступенчатом воздействии необходимо проделать следующие действия:

* 1. Умножим передаточную функцию на преобразование Лапласа ступенчатого воздействия, то есть на :
  2. Затем преобразуем полученное выражение к виду:
  3. Возьмём обратное преобразование Лапласа от полученного выражения и получим дифференциальное уравнение:
  4. Решив полученное дифференциальное уравнение получим функцию зависимости наблюдаемой величины от времени, что и будет являться переходной функцией системы при ступенчатом воздействии.

Среда MATLAB позволяет получать графики переходных функций, не проделывая всех этих действий. Для этого необходимо коэффициенты числителя и знаменателя подать на вход в функцию step(), а на выходе получаем график переходного процесса.

Полный код программы для первого пункта работы представлен в приложении 1.

Графики переходных процессов изображены на рисунках 2-7, а под каждым из них указаны прямые показатели качества этих процессов.

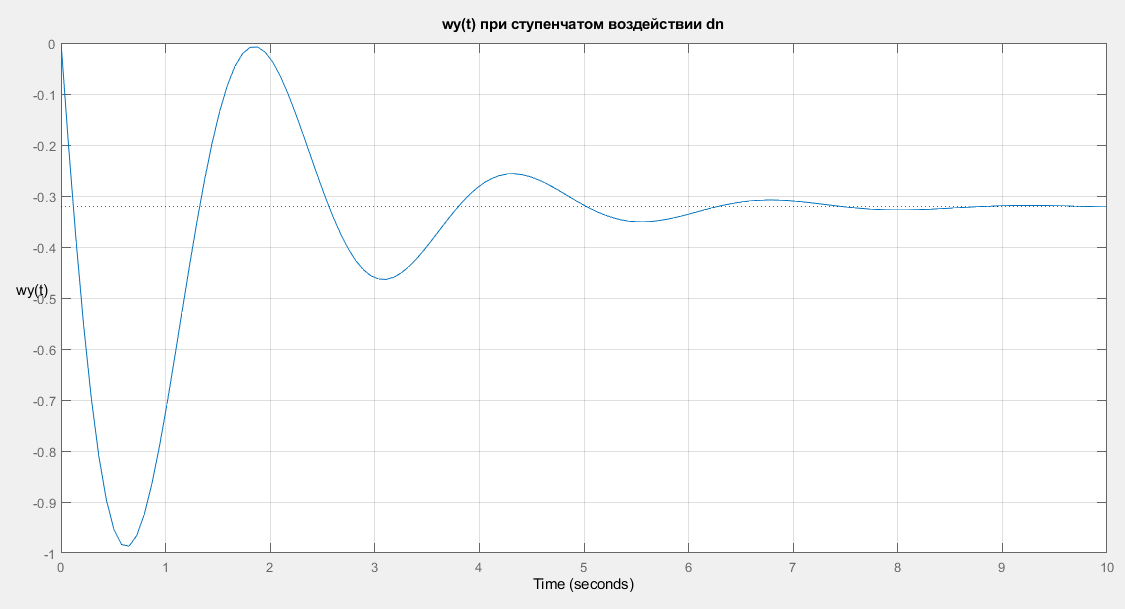


Рисунок 2. Переходный процесс изменения угловой скорости рысканья при ступенчатом воздействии рулей направления

Прямые показатели качества:

* перерегулирование:
* установившееся значение:
* время нарастания:
* время переходного процесса:
* колебательность:

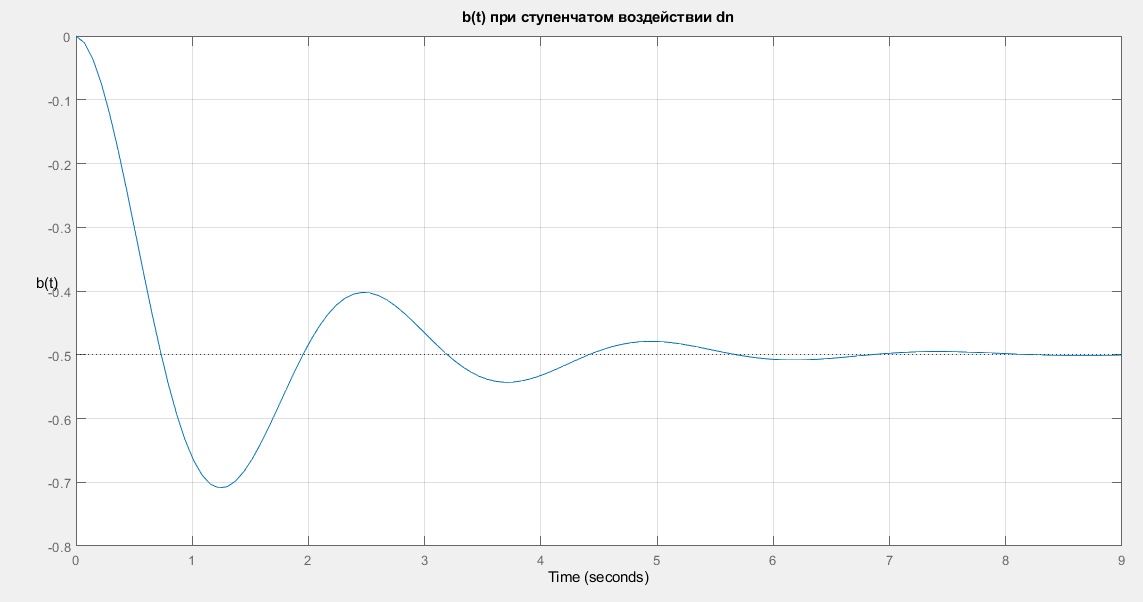


Рисунок 3. Переходный процесс изменения угла скольжения при ступенчатом воздействии рулей направления

Прямые показатели качества:

* перерегулирование:
* установившееся значение:
* время нарастания:
* время переходного процесса:
* колебательность:

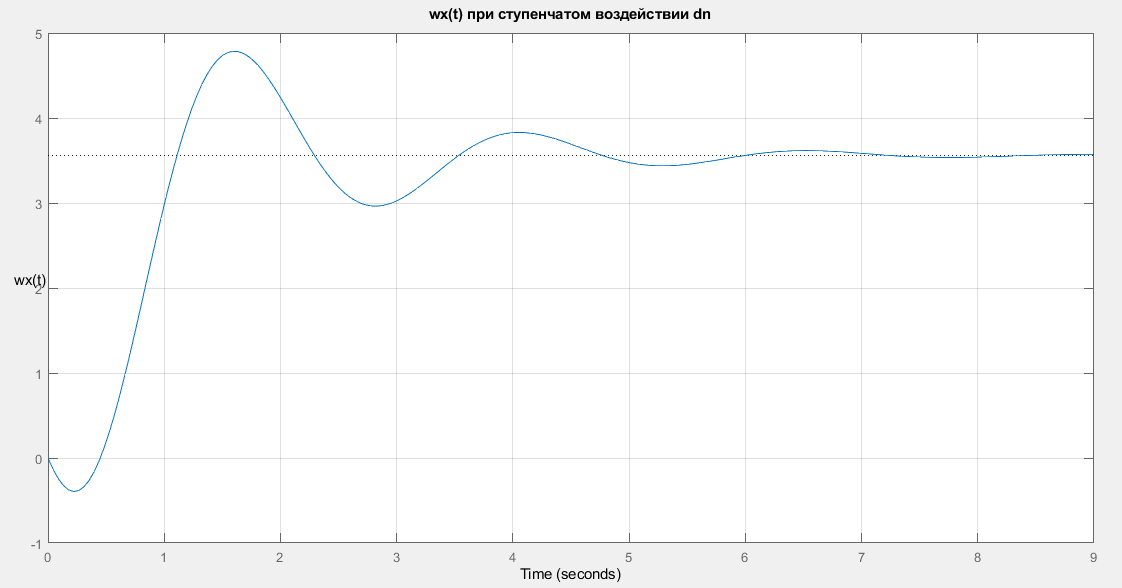


Рисунок 4. Переходный процесс изменения угловой скорости крена при ступенчатом воздействии рулей направления

Прямые показатели качества:

* перерегулирование:
* установившееся значение:
* время нарастания:
* время переходного процесса:
* колебательность:

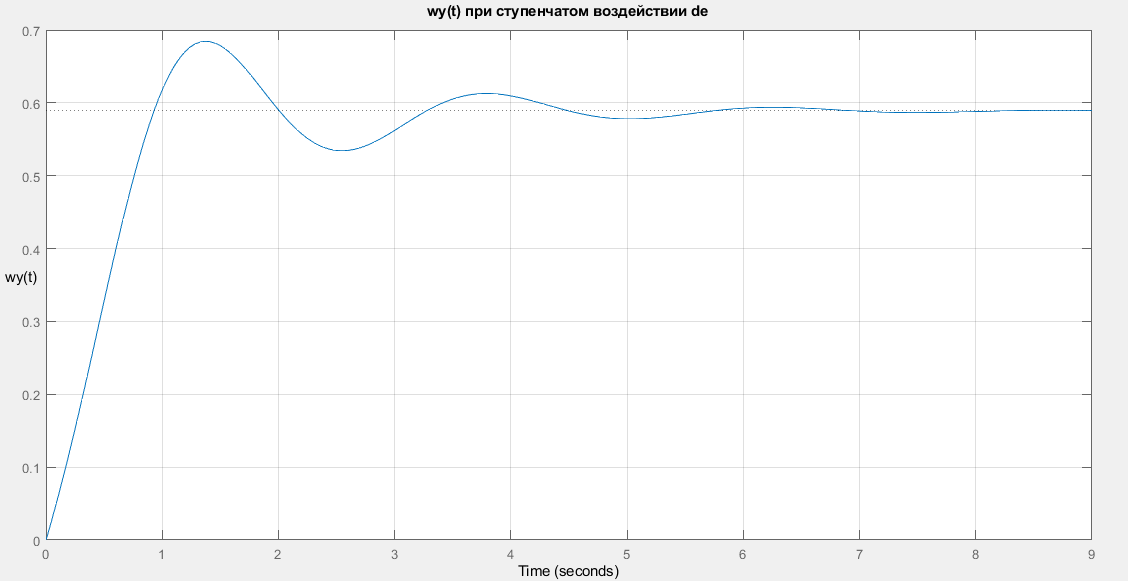


Рисунок 5. Переходный процесс изменения угловой скорости рысканья при ступенчатом воздействии элеронов

Прямые показатели качества:

* перерегулирование:
* установившееся значение:
* время нарастания:
* время переходного процесса:
* колебательность:

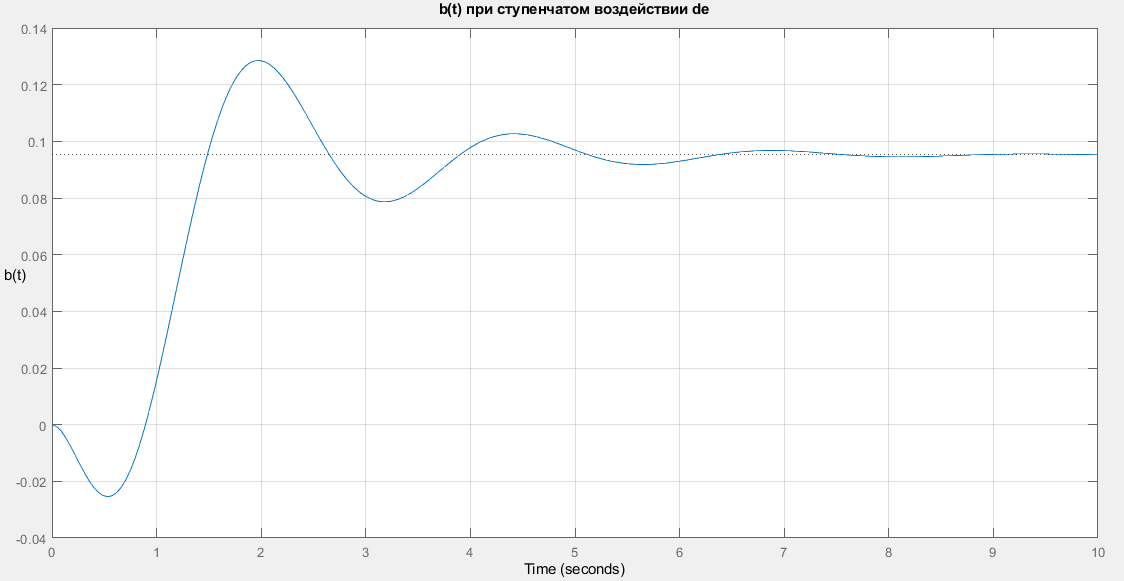


Рисунок 6. Переходный процесс изменения угла скольжения при ступенчатом воздействии элеронов

Прямые показатели качества:

* перерегулирование:
* установившееся значение:
* время нарастания:
* время переходного процесса:
* колебательность:

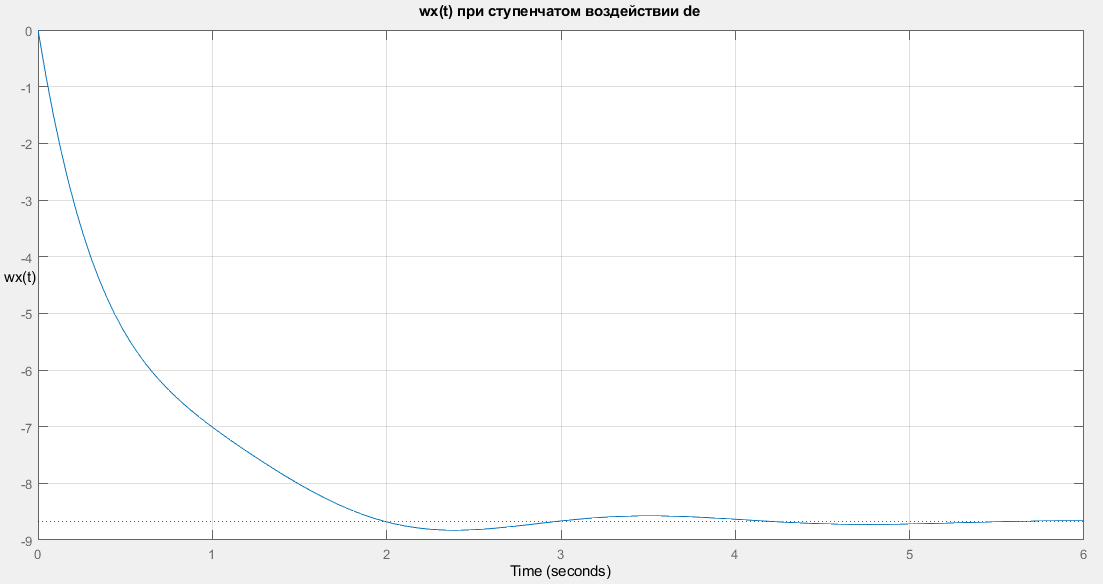


Рисунок 7. Переходный процесс изменения угловой скорости крена при ступенчатом воздействии элеронов

Прямые показатели качества:

* перерегулирование:
* установившееся значение:
* время нарастания:
* время переходного процесса:
* колебательность:

Также определим косвенные показатели качества.

Полюса передаточных функций (корни характеристического многочлена):

* степень устойчивости:
* степень колебательности:

Для улучшения управляемости в боковом канале необходимо произвести улучшение характеристик управляемости, так как заданная система не удовлетворяет требованиям: переходный процесс имеет большую степень колебательности в то время как он обязан иметь практически монотонный характер и иметь время переходного процесса не более чем 1 с. Также хорошо видно, что переходные процессы для других величин происходят относительно долго и сопровождаются большими колебаниями.

# **Выбор параметров цифрового закона управления в контуре демпфирования**

Рассмотрим упрощённую модель системы .

Тогда система 1 примет вид:

А из системы 2 можно будет исключить второе уравнение:

Тогда запишем общую систему уравнений упрощённой системы:

Можем преобразовать её к матричному виду:

Где

Тогда имеем

Где

– наблюдаемые величины.

После замыкания контура данную модель можно представить в виде следующей структурной схемы:

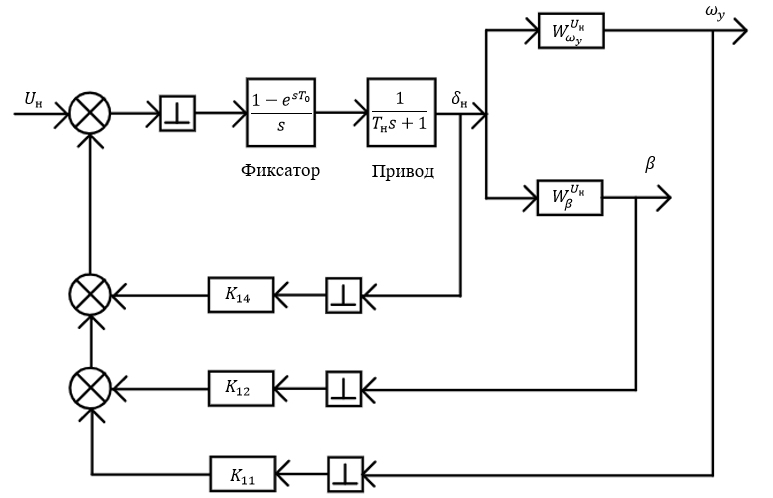


Рисунок 8. Структурная схема дискретной упрощённой системы управления боковым движением самолёта

Чтобы перейти от уравнений для непрерывной разомкнутой системы к уравнениям для дискретной разомкнутой системы вида

Необходимо вычислить матрицы перехода и .

Используя среду MATLABR2021b, вычислим данные матрицы с точностью до 20-го слагаемого:

Передаточные функции дискретной непрерывной разомкнутой системы:

Тогда можно представить данную систему в упрощённом виде

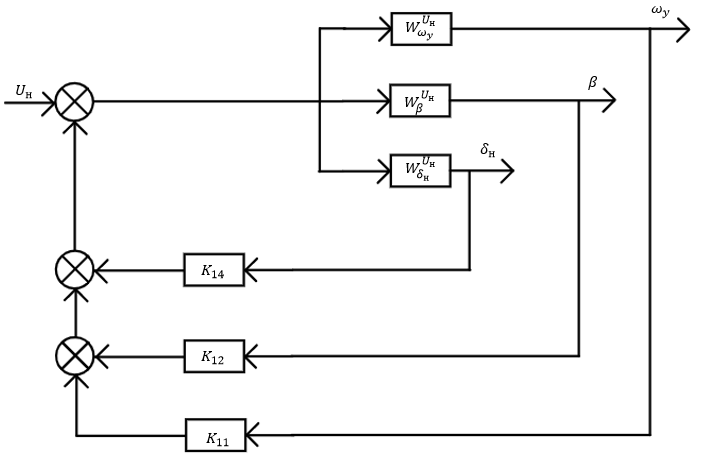


Рисунок 9. Упрощённый контур после преобразования

Передаточные функции замкнутой системы могут быть вычислены следующим образом:

Вычислим и получим следующие выражения для передаточных функций:

– характеристический многочлен системы.

Для того, чтобы выбрать необходимые коэффициенты обратной связи, зададимся некоторым номинальным характеристическим многочленом, полученным при сочетании колебательного и апериодического звена.

Рассмотрим характеристическое уравнение колебательного звена:

Тогда

Для того, чтобы удовлетворить требованиям системы

1. Необходимо, чтобы собственная частота колебаний была не менее 4.
2. Затухание колебаний не менее чем в 10 раз за период.

Тогда выберем , (𝜆 – взята с запасом).

Тогда получаем систему уравнений, из которой мы должны найти параметры колебательного звена, и .

Тогда получаем

Тогда передаточная функция с номинальным характеристическим многочленом будет выглядеть следующим образом:

Подставив найденные параметры и применив *Z-*преобразование к данной передаточной функции, получаем:

Где характеристический многочлен равен

Сравнивая характеристические многочлены исходной системы и номинальной передаточной функции, можно составить систему уравнений, решая которую, найдём все коэффициенты обратной связи.

Данную систему можно представить в виде матричного уравнения.

Решив матричное уравнение, получаем

Получим окончательно передаточные функции замкнутой системы, подставив полученные значения коэффициентов обратной связи:

Видно, что характеристический многочлен системы совпадает с номинальным.

Построим переходные процессы по всем каналам.

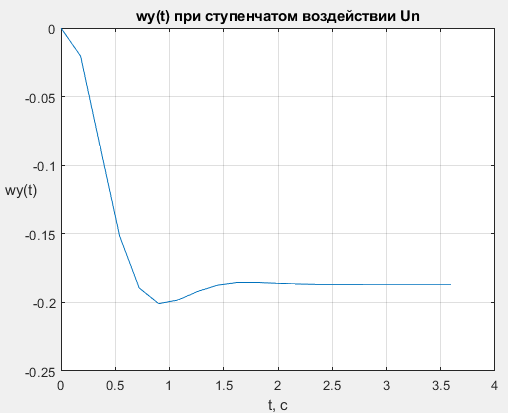


Рисунок 10. Переходный процесс изменения угловой скорости рысканья при ступенчатом воздействии в цифровом контуре

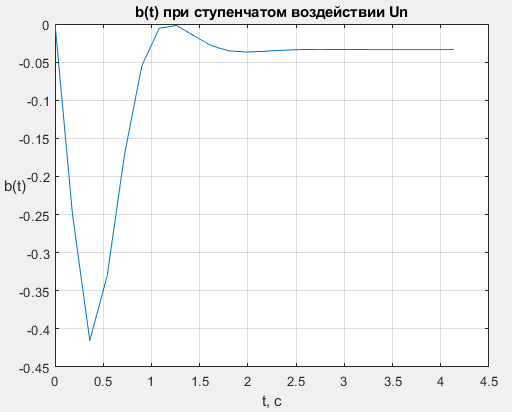


Рисунок 11. Переходный процесс изменения угла скольжения при ступенчатом воздействии в цифровом контуре

Переходные процессы удовлетворяют необходимым требованиям

Построим линейный наблюдатель, обеспечивающий измерение полного вектора состояния упрощённой системы по доступной координате .

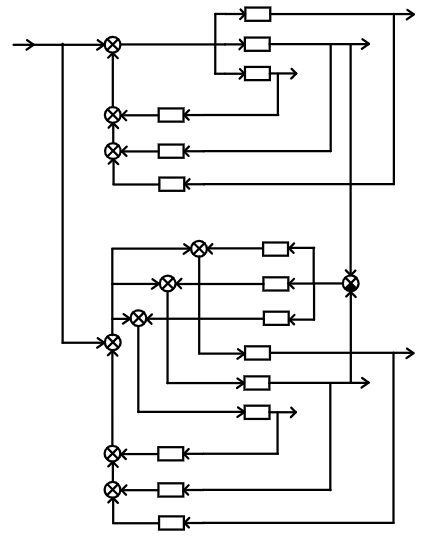


Рисунок 12. Структурная схема упрощённой системы с наблюдателем Калмана

Запишем дифференциальное уравнение объекта управления с замкнутой системой

Запишем дифференциальные уравнения наблюдателя Калмана с измеряемой координатой .

где

Так как – измеряемая координата, а и – нет, то

Необходимо найти коэффициенты наблюдателя Калмана:

Произведём необходимые преобразования

Тогда

Получаем дифференциальное уравнение ошибки.

Характеристический многочлен:

Тогда получаем

Зададим желаемый характеристический многочлен, соответствующий максимальной степени устойчивости. Тогда

При таких корнях время переходного процесса будет равно

Желаемое время переходного процесса

Тогда корни желаемого характеристического многочлена примут следующий вид:

Тогда желаемый характеристический многочлен примет следующий вид:

Или после z-преобразования:

Чтобы получить характеристический многочлен желаемого вида, сравним коэффициенты при соответствующих степенях характеристических многочленов и решим систему уравнений:

Получим

Теперь запишем систему уравнений обобщённой системы

Эту систему можно привести к обобщённому виду

Где

Построим переходные процессы для сравнения фактической выходной координаты и её оценки.

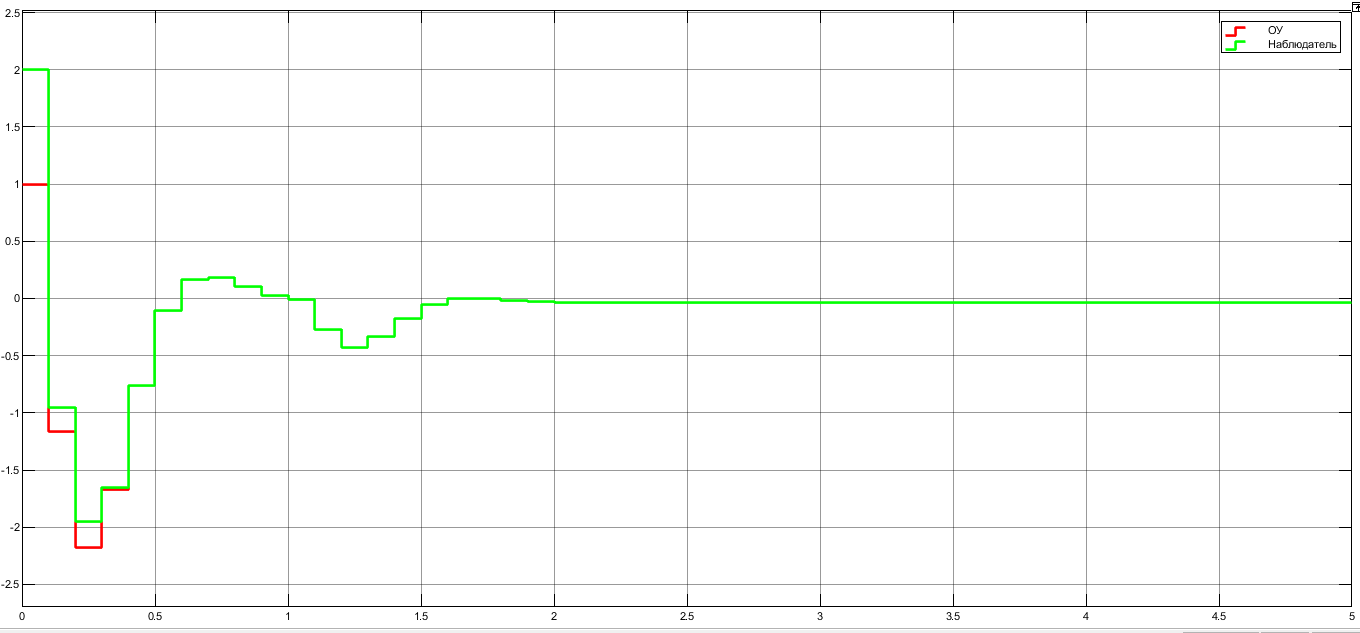


Рисунок 13. Переходный процесс по и

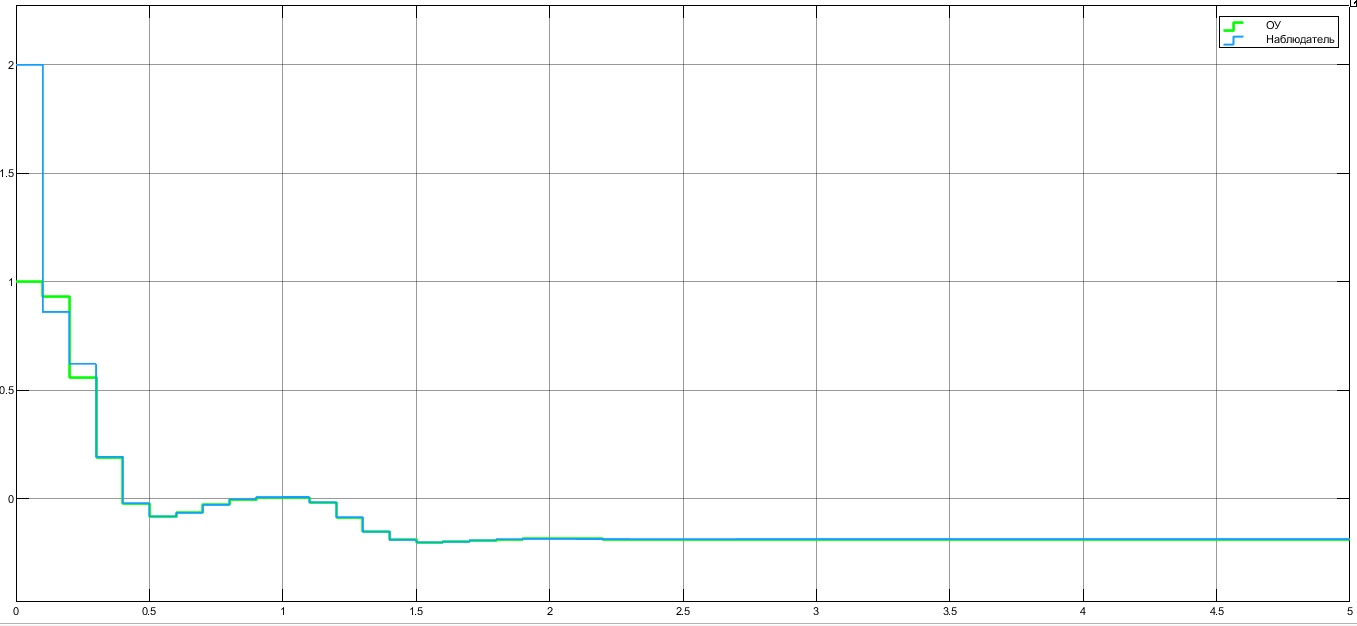


Рисунок 14. Переходный процесс по и

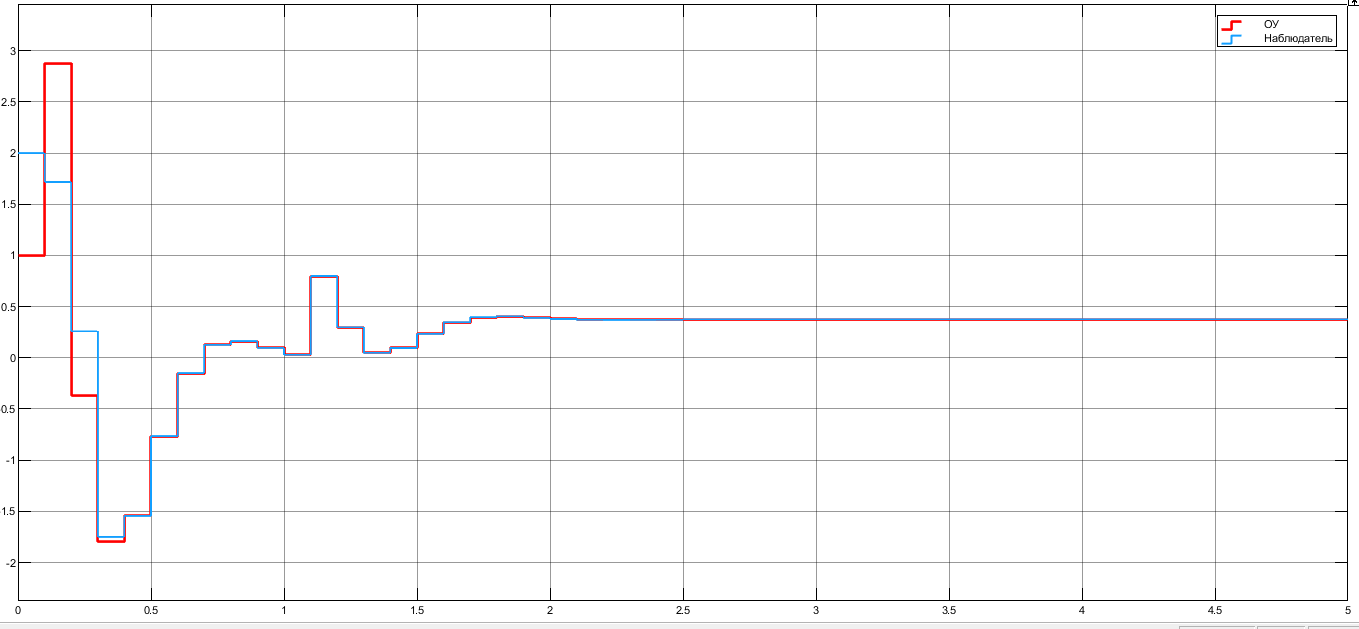


Рисунок 15. Переходный процесс по и

Как видно из графиков, наблюдаемое значение достаточно быстро сходится с реальным.

# **Построение области устойчивости упрощённой системы в плоскости параметров , .**

Пропустим этапы вычисления передаточных функций. Подобные расчёты проводились выше. В этом же пункте эти расчёты проводятся в программе, код которой приводится в приложении 3. В данном пункте – берётся из прошлого пункта.

Необходимо:

1. Составить матрицу передаточных функций цифровой системы замкнутого контура;
2. Заменить все на ;
3. Упростить полученные передаточные функции;
4. Выделить знаменатель передаточных функций;
5. Построить область устойчивости, используя критерий Гурвица.

Проделав первые 4 пункта, получим

Условия устойчивости для системы 3 порядка, согласно критерию Гурвица:

Построим границы устойчивости по следующему алгоритму:

1. Зададим сетку и с некоторым шагом;
2. Для каждого узла сетки проверяются условия системы 51;
3. Если условия выполняются, то узел считается принадлежащим области устойчивости, результат записывается в соответствующую матрицу в виде 1, а иначе – в виде 0.
4. Получившаяся матрица булевых значений записывается в виде графика, соответствующего области устойчивости (рисунок 16).

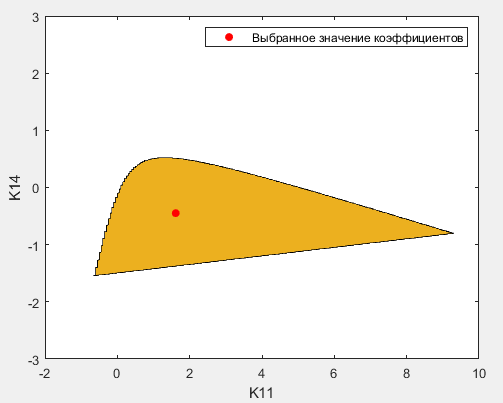


Рисунок 16. Область устойчивости замкнутой системы по коэффициентам и

Для построения области устойчивости, соответствующей двукратному запасу устойчивости, в системе 64 все коэффициенты обратной связи увеличиваются в 2 раза (рисунок 17).

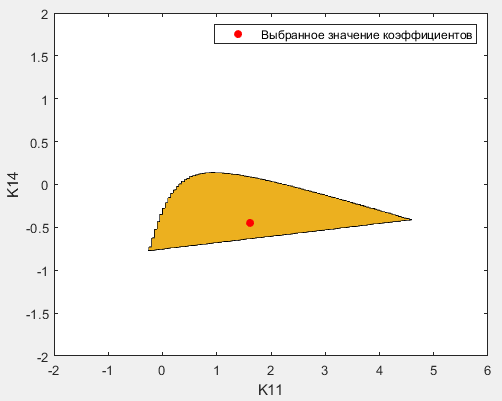


Рисунок 17. Область устойчивости замкнутой системы, соответствующая двукратному запасу устойчивости

Видно, что полученные в пункте 2 коэффициенты лежат внутри области устойчивости, следовательно, запас устойчивости при данных коэффициентах больше 2.

# **Выбор параметров закона управления в канале элеронов**

Для того, чтобы переходный процесс по удовлетворял требованиям, значение коэффициента обратной связи можно подобрать, используя ЛАФЧХ.

Определим матрицы системы замкнутой по и , но разомкнутой по :





Вектор состояния системы и управляющий вектор:

Получим передаточную функцию разомкнутой системы по в канале элеронов разомкнутой системы:

Выполним билинейное преобразования, а затем перейдём в пространство 𝜆.

Построим ЛАФЧХ разомкнутой системы:

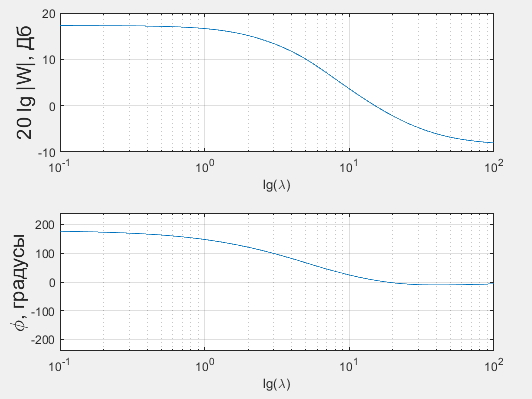


Рисунок 18. ЛАФЧХ для разомкнутой системы по при управлении по

Выберем коэффициент обратной связи по равный

Тогда матрица , коэффициентов обратной связи, будет иметь следующий вид



Проделав все те же самые действия, что и ранее, построим ЛАФЧХ замкнутой системы.

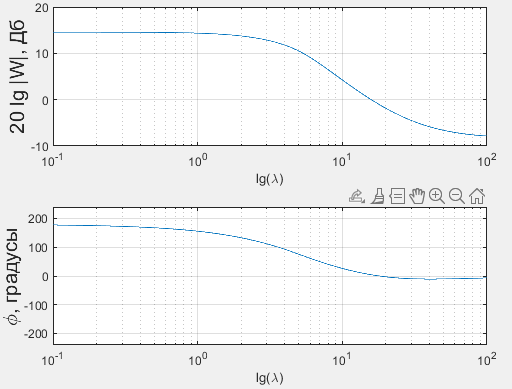


Рисунок 19. ЛАФЧХ для замкнутой системы по при управлении по

Также построим переходный процесс по .

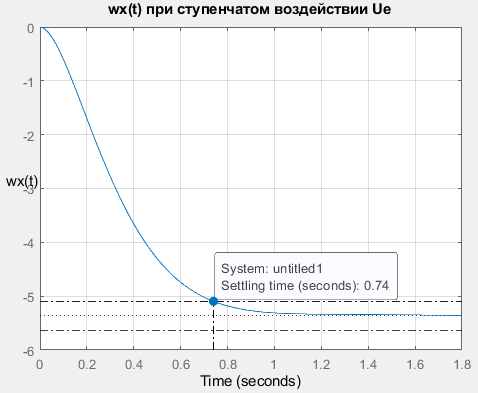


Рисунок 20. Переходный процесс в канале рысканья в полной замкнутой системе

Переходный процесс удовлетворяет необходимым требованиям, т.к. время переходного процесса не превышает 1 с, а перерегулированием не превышает 5%.

# **Выбор коэффициента электрической связи от ручки лётчика**

Расход ручки на приращение в установившемся режиме управления по должен соответствовать величине .

Значение в установившемся режиме .

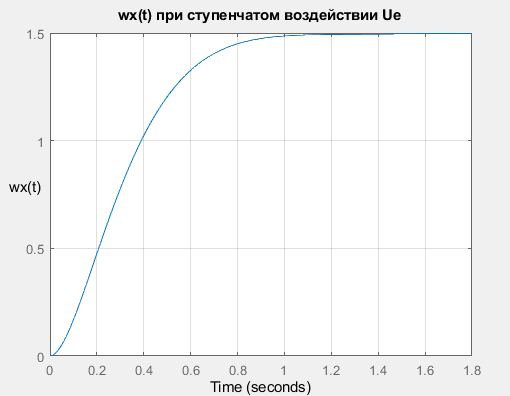
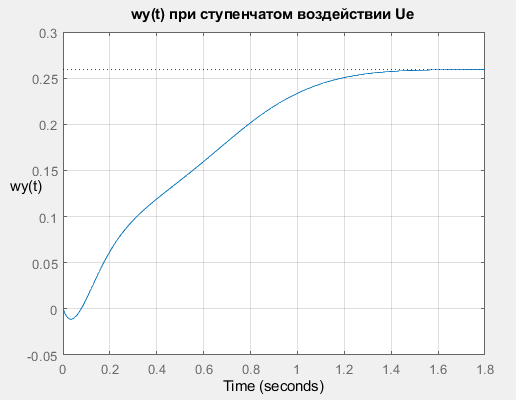
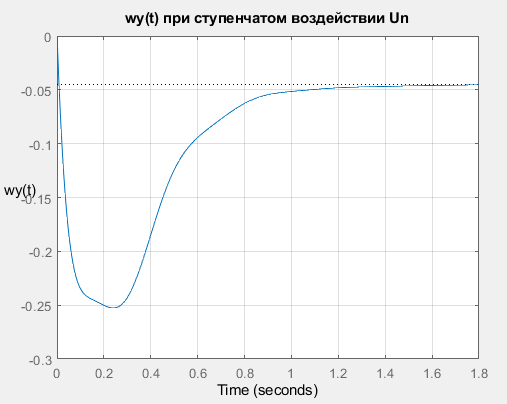
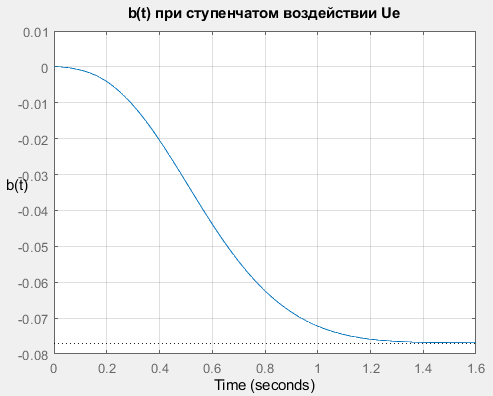
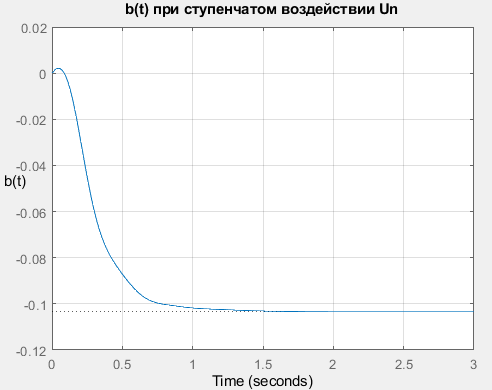


Рисунок 21. Переходный процесс по ωx при отклонении ручки на 1 мм.

# **Проверка заданных требований**

Увеличим значения всех коэффициентов обратной связи в 2 раза, чтобы проверить, что система удовлетворяет двукратному запасу устойчивости и построим переходные процессы.





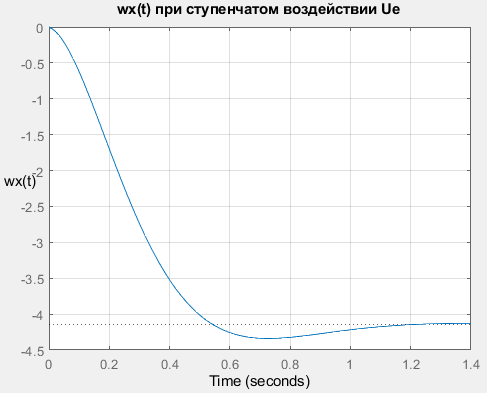
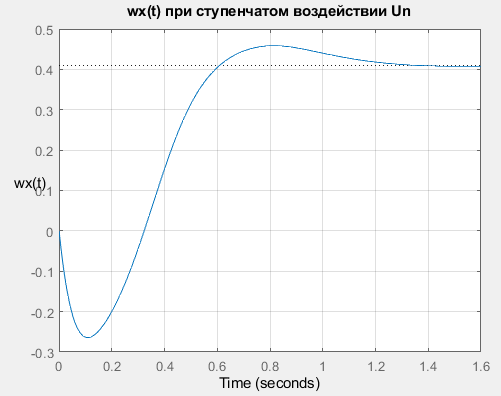
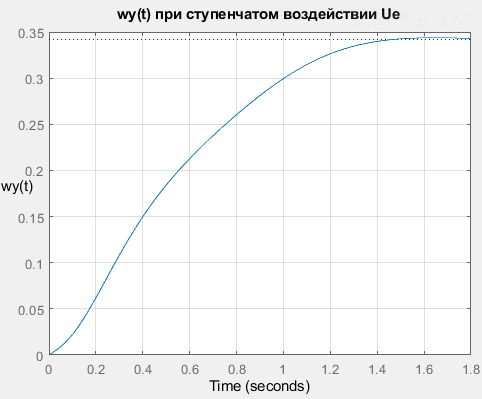
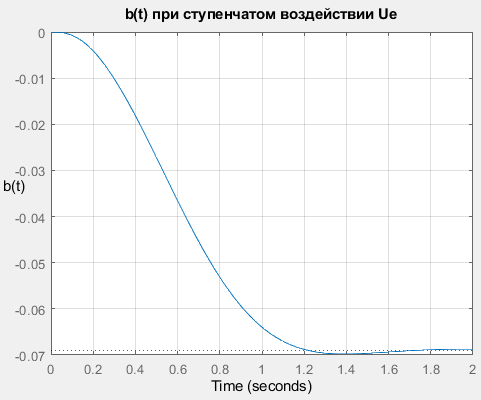
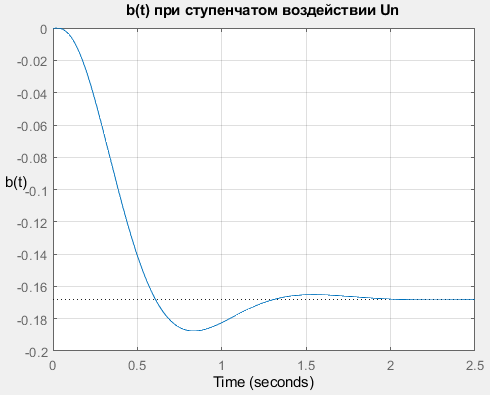


Рисунок 22. Переходные процессы при удвоенных коэффициентах обратных связей

Все процессы устойчивы, значит система обладает двукратным запасом устойчивости по всем коэффициентам.





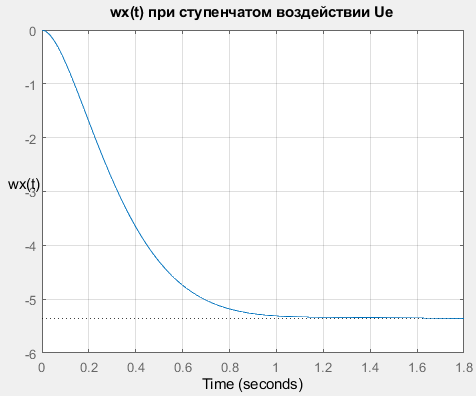
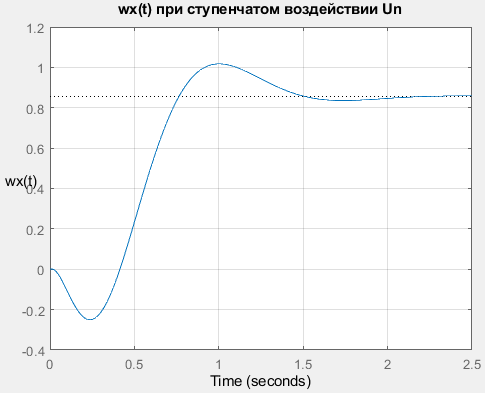


Рисунок 23. Переходные процессы при выбранных коэффициентах обратной связи

Переходный процесс по *ωx* при управлении по элеронам имеет время переходного процесса менее 1 секунды и перерегулирование менее 5%. Затухание колебаний по и при управлении по рулям направления более чем в 10 раз за период, т.е. система удовлетворяет всем необходимым требованиям.

# **Полный алгоритм управления боковым движением**

Полная система с цифровым алгоритмом управления может быть описана схемой, изображённой на рисунке 19. На вход цифрового управляющего блока поступают 6 сигналов: , . Выходами являются управляющие сигналы: .

Обозначим используемые переменные для всех величин и коэффициентов

Таблица 1. Таблица значений и переменных алгоритма управления

|  |  |
| --- | --- |
| Значение переменной | Имя переменной |
|  | X1 |
|  | X2 |
|  | X3 |
|  | X4 |
|  | X5 |
|  | X6 |
|  | X7 |
|  | Y1 |
|  | Y2 |
|  | C1 |
|  | C2 |
|  | C3 |
|  | C4 |
|  | C5 |
|  | C6 |
| Промежуточная переменная для умножения | P |
| Промежуточная переменная для сложения | S |

Описание алгоритма:

1. Поместить входное значение в переменную X1.
2. Поместить входное значение *β* в переменную X2.
3. Поместить входное значение в переменную X3.
4. Поместить входное значение в переменную X4.
5. Поместить входное значение в переменную X5.
6. Поместить входное значение в переменную X6.
7. Поместить входное значение в переменную X7.
8. Умножить значение переменной X1 на значение константы С1, результат присвоить переменной *P*.
9. Присвоить значение переменной *P* переменной *S*, т.e.
10. Умножить значение переменной X2 на значение константы С2, результат присвоить переменной *P*. .
11. Добавить к значению переменной *S* содержимое переменной *P*, т.е.
12. Умножить значение переменной X4 на значение константы С3, результат присвоить переменной *P*. .
13. Добавить к значению переменной *S* содержимое переменной *P*, т.е.
14. Умножить значение переменной X7 на значение константы С6, результат присвоить переменной *P*. .
15. Добавить к значению переменной *S* содержимое переменной *P*, т.е.
16. Переместить значение ячейки *S* в переменную Y1.
17. Умножить значение переменной X3 на значение константы С4, результат присвоить переменной *P*. .
18. Присвоить переменной *S* значение переменной *P*, т.е.
19. Умножить значение переменной X6 на значение константы С5, результат присвоить переменной *P*. .
20. Добавить к значению переменной *S* содержимое переменной *P*, т.е.
21. Переместить значение ячейки *S* в переменную Y2.

По приведенному выше алгоритму понадобится:

* 7 переменных для входных значений;
* 6 переменных для констант;
* 2 переменные для выходных значений;
* 2 промежуточные переменные;
* 6 операций умножения;
* 4 операций сложения;
* 27 операций;

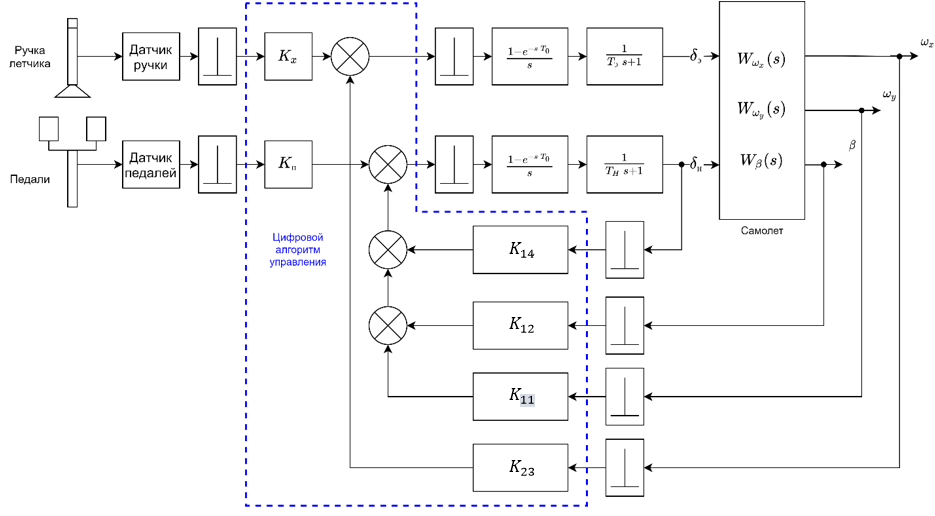


Рисунок 24. Полная схема управления

# **Заключение**

В ходе курсовой работы была исследована динамика САУ ЛА самолётом, а также синтезирован цифровой алгоритм управления.

В процессе исследования динамики было выяснено, что система удовлетворяет техническим требованиям.

# **Список иллюстраций**

[Рисунок 1. Структурная схема цифровой системы управления 5](#_Toc103921588)

[Рисунок 2. Переходный процесс изменения угловой скорости рысканья при ступенчатом воздействии рулей направления 9](#_Toc103921589)

[Рисунок 3. Переходный процесс изменения угла скольжения при ступенчатом воздействии рулей направления 10](#_Toc103921590)

[Рисунок 4. Переходный процесс изменения угловой скорости крена при ступенчатом воздействии рулей направления 11](#_Toc103921591)

[Рисунок 5. Переходный процесс изменения угловой скорости рысканья при ступенчатом воздействии элеронов 12](#_Toc103921592)

[Рисунок 6. Переходный процесс изменения угла скольжения при ступенчатом воздействии элеронов 13](#_Toc103921593)

[Рисунок 7. Переходный процесс изменения угловой скорости крена при ступенчатом воздействии элеронов 14](#_Toc103921594)

[Рисунок 8. Структурная схема дискретной упрощённой системы управления боковым движением самолёта 17](#_Toc103921595)

[Рисунок 9. Упрощённый контур после преобразования 19](#_Toc103921596)

[Рисунок 10. Переходный процесс изменения угловой скорости рысканья при ступенчатом воздействии в цифровом контуре 23](#_Toc103921597)

[Рисунок 11. Переходный процесс изменения угла скольжения при ступенчатом воздействии в цифровом контуре 23](#_Toc103921598)

[Рисунок 12. Структурная схема упрощённой системы с наблюдателем Калмана 24](#_Toc103921599)

[Рисунок 13. Переходный процесс по и 28](#_Toc103921600)

[Рисунок 14. Переходный процесс по и 29](#_Toc103921601)

[Рисунок 15. Переходный процесс по и 29](#_Toc103921602)

[Рисунок 16. Область устойчивости замкнутой системы по коэффициентам и 31](#_Toc103921603)

[Рисунок 17. Область устойчивости замкнутой системы, соответствующая двукратному запасу устойчивости 32](#_Toc103921604)

[Рисунок 18. ЛАФЧХ для разомкнутой системы по при управлении по 34](#_Toc103921605)

[Рисунок 19. ЛАФЧХ для замкнутой системы по при управлении по 35](#_Toc103921606)

[Рисунок 20. Переходный процесс в канале рысканья в полной замкнутой системе 36](#_Toc103921607)

[Рисунок 21. Переходный процесс по ωx при отклонении ручки на 1 мм. 37](#_Toc103921608)

[Рисунок 22. Переходные процессы при удвоенных коэффициентах обратных связей 38](#_Toc103921609)

[Рисунок 23. Переходные процессы при выбранных коэффициентах обратной связи 39](#_Toc103921610)

[Рисунок 24. Полная схема управления 44](#_Toc103921611)

# **Список использованных источников**

1. ПОСТРОЕНИЕ МАТРИЦЫ ПЕРЕДАТОЧНЫХ ФУНКЦИЙ МНОГОМЕРНОЙ СИСТЕМЫ С ПОМОЩЬЮ ПОТОЧЕЧНОГО РАСЧЕТА ОПРЕДЕЛИТЕЛЕЙ. 1996 г. А. В. ЗЕМСКОВ, канд. техн. наук (Саратовское высшее военное командно-инженерное училище ракетных войск имени А. И. Лизюкова). Методические указания.
2. Белоногов Виктор Дмитриевич – Лекции по курсу цифровых систем управления 2022.

# **Приложение 1**

clc

clear

syms s

% Зададим матрицы системы уравнений

A = [-0.6, -5.71, -0.04;...

1, -0.26, 0.065;...

-0.7, -24, -2.5];

B = [-2.9, 0.55;...

-0.04, 0;...

-3.3, -19];

C = [1, 0, 0;...

0, 1, 0;...

0, 0, 1];

D = 0;

I = eye(3, 3);

% Получаем матрицу передаточных функций от всех управляющих воздействий ко

% всем наблюдаемым величинам

W\_matrix = simplify(C \* inv((s\*I - A)) \* B);

% Выводим передаточную функцию W\_wx\_dn в стандартном виде отношения полиномов

W\_wy\_dn = W\_matrix(1, 1);

W\_wy\_dn = convert\_to\_tf(W\_wy\_dn, false)

% Выводим передаточную функцию W\_b\_dn в стандартном виде отношения полиномов

W\_b\_dn = W\_matrix(2, 1);

W\_b\_dn = convert\_to\_tf(W\_b\_dn, false)

% Выводим передаточную функцию W\_wy\_dn в стандартном виде отношения полиномов

W\_wx\_dn = W\_matrix(3, 1);

W\_wx\_dn = convert\_to\_tf(W\_wx\_dn, false)

% Выводим передаточную функцию W\_wx\_de в стандартном виде отношения полиномов

W\_wy\_de = W\_matrix(1, 2);

W\_wy\_de = convert\_to\_tf(W\_wy\_de, false)

% Выводим передаточную функцию W\_b\_de в стандартном виде отношения полиномов

W\_b\_de = W\_matrix(2, 2);

W\_b\_de = convert\_to\_tf(W\_b\_de, false)

% Достаём передаточную функцию W\_wy\_de в стандартном виде отношения полиномов

W\_wx\_de = W\_matrix(3, 2);

W\_wx\_de = convert\_to\_tf(W\_wx\_de, false)

% Строим графики при помощи созданной нами функции tf\_and\_step(W)

figure;

step(W\_wy\_dn);

grid on;

title('wy(t) при ступенчатом воздействии dn');

ylabel('wy(t)', Rotation=0);

figure;

step(W\_b\_dn);

grid on;

title('b(t) при ступенчатом воздействии dn');

ylabel('b(t)', Rotation=0);

figure;

step(W\_wx\_dn);

grid on;

title('wx(t) при ступенчатом воздействии dn');

ylabel('wx(t)', Rotation=0);

figure;

step(W\_wy\_de);

grid on;

title('wy(t) при ступенчатом воздействии de');

ylabel('wy(t)', Rotation=0);

figure;

step(W\_b\_de);

grid on;

title('b(t) при ступенчатом воздействии de');

ylabel('b(t)', Rotation=0);

figure;

step(W\_wx\_de);

grid on;

title('wx(t) при ступенчатом воздействии de');

ylabel('wx(t)', Rotation=0);

% Получаем полюса передаточных функций при помощи созданной нами функции

% POLUSES(W)

disp([newline, 'Полюса (корни характеристического многочлена)'])

POLUSES\_W\_wx\_dn = roots(sym2poly(det(s\*I - A)))

Программа 1. Основной код для 1 пункта задания.

function [outputArg] = convert\_to\_tf(W\_in, is\_z\_func)

% Преобразование дроби к типу данных tf

T0 = 0.18;

[a, b] = numden(W\_in);

a = sym2poly(a);

b = sym2poly(b);

a = a / b(end);

b = b / b(end);

if is\_z\_func

W\_out = tf(a, b, T0);

else

W\_out = tf(a, b);

end

outputArg = W\_out;

end

Программа 2. Функция для построения переходного процесса с возвращаемым значением передаточной функции

# **Приложение 2**

clc

clear

syms s z K11 K12 K14

% Зададим постоянную времени

Tn = 0.11;

% Зададим время дискретизации

T0 = 0.18;

% Зададим матрицы системы уравнений

A = [-0.6, -5.71, -2.9;...

1, -0.26, -0.04;...

0, 0, -1/Tn];

B = [0;...

0;...

1/Tn];

C = [1, 0, 0;...

0, 1, 0;...

0, 0, 1];

D = 0;

I = eye(3, 3);

% Вычислим матрицу перехода G

G = 0;

n = 20;

for i = 0:n

G = G + A^i \* T0^i / factorial(i);

end

% Вычислим матрицу перехода Dd

Dd = 0;

for i = 0:n

Dd = Dd + A^(i) \* T0^(i+1) / factorial(i+1);

end

Dd = Dd \* B;

% Вычислим матрицу передаточных функций разомкнутой цифровой системы

Wz = collect((inv(z\*I-G))\*Dd);

% Выделим передаточные функции по всем каналам

W\_omy\_Un = collect(Wz(1));

W\_b\_Un = collect(Wz(2));

W\_sigmn\_Un = collect(Wz(3));

% Преобразуем ПФ к типу tf

tf\_W\_omy\_Un = convert\_to\_tf(Wz(1), true);

tf\_W\_b\_Un = convert\_to\_tf(Wz(2), true);

tf\_W\_sigmn\_Un = convert\_to\_tf(Wz(3), true);

% Вычислим передаточные функции замкнутого цифрового контура и приведём их

% к нормальному виду

W\_omy\_Un\_z = collect(W\_omy\_Un / (1 - K11\*W\_omy\_Un - K12\*W\_b\_Un - K14\*W\_sigmn\_Un));

[a, b] = numden(W\_omy\_Un\_z);

c = coeffs(b, z);

c = c(end);

a\_omy\_Un\_z = vpa(collect(a / c));

b\_omy\_Un\_z = vpa(collect(b / c));

W\_b\_Un\_z = collect(W\_b\_Un / (1 - K11\*W\_omy\_Un - K12\*W\_b\_Un - K14\*W\_sigmn\_Un));

[a, b] = numden(W\_b\_Un\_z);

c = coeffs(b, z);

c = c(end);

a\_b\_Un\_z = vpa(collect(a / c));

b\_b\_Un\_z = vpa(collect(b / c));

W\_sigmn\_Un\_z = collect(W\_sigmn\_Un / (1 - K11\*W\_omy\_Un - K12\*W\_b\_Un - K14\*W\_sigmn\_Un));

[a, b] = numden(W\_sigmn\_Un\_z);

c = coeffs(b, z);

c = c(end);

a\_sigmn\_Un\_z = vpa(collect(a / c));

b\_sigmn\_Un\_z = vpa(collect(b / c));

% Вычислим номинальную передаточную функцию, исходя из требований

T = sqrt(1 / ( 16 + (log(20))^2 ));

psi = log(20) \* T;

Wn = (1 / (T^2\*Tn\*s^3 + (2\*Tn\*psi\*T + T^2)\*s^2 + (Tn + 2\*T\*psi)\*s + 1));

Wn = convert\_to\_tf(Wn, false);

Wn\_z = c2d(Wn, T0);

% Найдём коэффициенты обратной связи, проводя сравнение коэффициентов с

% номинальным характеристическим многочленом

A\_K = [0.24877, 0.02045, -0.80531;...

-0.10019, 0.04396, 1.35556;...

-0.13607, 0.00514, -0.68982];

B\_K = [0.80596;...

-0.67349;...

0.10054];

K = inv(A\_K)\*B\_K;

K = K';

% Подставим полученные значения коэффициентов в передаточные функции

% замкнутого цифрового контура

b\_omy\_Un\_z = subs(b\_omy\_Un\_z, [K11, K12, K14], [K(1), K(2), K(3)]);

b\_b\_Un\_z = subs(b\_b\_Un\_z, [K11, K12, K14], [K(1), K(2), K(3)]);

b\_sigmn\_Un\_z = subs(b\_sigmn\_Un\_z, [K11, K12, K14], [K(1), K(2), K(3)]);

% Приведём всё к типу данным tf

a\_omy\_Un\_z = sym2poly(a\_omy\_Un\_z);

b\_omy\_Un\_z = sym2poly(b\_omy\_Un\_z);

W\_omy\_Un\_z = tf(a\_omy\_Un\_z, b\_omy\_Un\_z, T0);

a\_b\_Un\_z = sym2poly(a\_b\_Un\_z);

b\_b\_Un\_z = sym2poly(b\_b\_Un\_z);

W\_b\_Un\_z = tf(a\_b\_Un\_z, b\_b\_Un\_z, T0);

a\_sigmn\_Un\_z = sym2poly(a\_sigmn\_Un\_z);

b\_sigmn\_Un\_z = sym2poly(b\_sigmn\_Un\_z);

W\_sigmn\_Un\_z = tf(a\_sigmn\_Un\_z, b\_sigmn\_Un\_z, T0);

% Построим графики переходных функций

d\_step(W\_b\_Un\_z)

grid on;

title('wy(t) при ступенчатом воздействии Un');

ylabel('wy(t)', Rotation=0);

d\_step(W\_omy\_Un\_z)

grid on;

title('b(t) при ступенчатом воздействии Un');

ylabel('b(t)', Rotation=0);

d\_step(W\_sigmn\_Un\_z)

grid on;

title('dn(t) при ступенчатом воздействии Un');

ylabel('dn(t)', Rotation=0);

Программа 3. Основной код для пункта 2

clc

clear

syms s z L1 L2 L4 lambd

% Зададим постоянную времени

Tn = 0.11;

% Зададим время дискретизации

T0 = 0.18;

% Зададим матрицы системы уравнений

A = [-0.6, -5.71, -2.9;...

1, -0.26, -0.04;...

0, 0, -1/Tn];

B = [0;...

0;...

1/Tn];

C = [0, 1, 0];

D = 0;

% Вычислим матрицу перехода G

G = 0;

n = 20;

for i = 0:n

G = G + A^i \* T0^i / factorial(i);

end

% Вычислим матрицу перехода Dd

Dd = 0;

for i = 0:n

Dd = Dd + A^(i) \* T0^(i+1) / factorial(i+1);

end

Dd = Dd \* B;

I = eye(3, 3);

K = [1.6127, 2.1617, -0.4477];

L = [L1;...

L2;...

L4];

% Вычислим характеристический многочлен замкнутого фильтра Калмана

Dz = vpa(collect(det(lambd\*I - (G + Dd\*K - L\*C)), lambd));

coef = coeffs(Dz, lambd);

a0 = coef(1);

a1 = coef(2);

a2 = coef(3);

a3 = coef(4);

% Получим желаемый характеристический многочлен

tg = 0.3;

tn = 6.3;

t = tg / tn;

Dg = vpa(collect((lambd + 1/t)^3));

TF\_DG = convert\_to\_tf(1/Dg, false);

TFd\_DG = c2d(TF\_DG, T0);

b0 = TFd\_DG.denominator{1}(4);

b1 = TFd\_DG.denominator{1}(3);

b2 = TFd\_DG.denominator{1}(2);

b3 = TFd\_DG.denominator{1}(1);

eq0 = a0 == b0;

eq1 = a1 == b1;

eq2 = a2 == b2;

root = struct2array(solve(eq0, eq1, eq2, L1, L2, L4));

L(1) = root(1);

L(2) = root(2);

L(3) = root(3);

L = double(L);

Q = [G + Dd\*K, zeros(3, 3);...

L\*C, G+Dd\*K-L\*C];

R = [1, 0, 0, 0, 0, 0;

0, 1, 0, 0, 0, 0;

0, 0, 1, 0, 0, 0;

0, 0, 0, 1, 0, 0;

0, 0, 0, 0, 1, 0;

0, 0, 0, 0, 0, 1];

P = [Dd;

Dd];

W = inv(z\*eye(6) - Q) \* P;

W\_1 = convert\_to\_tf(vpa(collect(W(1))), true)

W\_2 = convert\_to\_tf(vpa(collect(W(2))), true)

W\_3 = convert\_to\_tf(vpa(collect(W(3))), true)

W\_4 = convert\_to\_tf(vpa(collect(W(4))), true)

W\_5 = convert\_to\_tf(vpa(collect(W(5))), true)

W\_6 = convert\_to\_tf(vpa(collect(W(6))), true)

figure;

step(W\_1, W\_4)

figure;

step(W\_2, W\_5)

figure;

step(W\_3, W\_6)

Программа 4. Код для расчёта наблюдателя

function [outputArg] = d\_step(W)

% Построение графика переходного процесса по заданной передаточной функции

figure;

step(W);

grid on;

xlabel('t, c');

ylabel('h(t)', Rotation=0);

outputArg = W;

end

Программа 5. Функция для построения графиков

function [outputArg] = convert\_to\_tf(W\_in, is\_z\_func)

% Преобразование дроби к типу данных tf

T0 = 0.18;

[a, b] = numden(W\_in);

a = sym2poly(a);

b = sym2poly(b);

a = a / b(end);

b = b / b(end);

if is\_z\_func

W\_out = tf(a, b, T0);

else

W\_out = tf(a, b);

end

outputArg = W\_out;

end

Программа 6. Функция для преобразования дроби к типу данных tf

# **Приложение 3**

clc

clear

syms s z K11 K12 K14 w

% Зададим постоянную времени

Tn = 0.11;

% Зададим время дискретизации

T0 = 0.18;

% Зададим матрицы системы уравнений

A = [-0.6, -5.71, -2.9;...

1, -0.26, -0.04;...

0, 0, -1/Tn];

B = [0;...

0;...

1/Tn];

C = [0, 1, 0];

D = 0;

% Вычислим матрицу перехода G

G = 0;

n = 20;

for i = 0:n

G = G + A^i \* T0^i / factorial(i);

end

% Вычислим матрицу перехода Dd

Dd = 0;

for i = 0:n

Dd = Dd + A^(i) \* T0^(i+1) / factorial(i+1);

end

Dd = Dd \* B;

I = eye(3, 3);

% Запишем коэффициенты обратной связи

K12 = 2.1617;

K = [K11, 2.1617, K14];

% Характеристический многочлен системы

% Вычислим матрицу передаточных функций разомкнутой цифровой системы

W = (inv((z\*I - (G + Dd\*K))))\*Dd;

% Выделим передаточные функции по всем каналам

W\_omy\_Un\_z = collect(W(1));

W\_b\_Un\_z = collect(W(2));

W\_sigmn\_Un\_z = collect(W(3));

% Заменим во всех передаточных функциях z на (1+w)/(1-w)

W\_omy\_Un\_w = subs(W\_omy\_Un\_z, z, (1+w)/(1-w));

[a, b] = numden(W\_omy\_Un\_w);

c = max(coeffs(b));

a\_omy\_Un\_w = vpa(collect(a / c));

b\_omy\_Un\_w = vpa(collect(b / c));

W\_b\_Un\_w = subs(W\_b\_Un\_z, z, (1+w)/(1-w));

[a, b] = numden(W\_b\_Un\_w);

c = max(coeffs(b));

a\_b\_Un\_w = vpa(collect(a / c));

b\_b\_Un\_w = vpa(collect(b / c));

W\_sigmn\_Un\_w = subs(W\_sigmn\_Un\_z, z, (1+w)/(1-w));

[a, b] = numden(W\_sigmn\_Un\_w);

c = max(coeffs(b));

a\_sigmn\_Un\_w = vpa(collect(a / c));

b\_sigmn\_Un\_w = vpa(collect(b / c));

% Вычислим характеристический многочлен и выделим коэффициенты

raus\_znam = b\_b\_Un\_w;

coef\_raus = coeffs(raus\_znam, w);

a0 = coef\_raus(1);

a1 = coef\_raus(2);

a2 = coef\_raus(3);

a3 = coef\_raus(4);

% Запишем неравенства ждя критерия Рауса

eq1 = a0 > 0;

eq2 = a1 > 0;

eq3 = a2 > 0;

eq4 = a3 > 0;

eq5 = simplify(a1\*a2 - a0\*a3) > 0;

% Преобразуем неравенства из символьного типа данных в функции

eq1 = matlabFunction(eq1);

eq2 = matlabFunction(eq2);

eq3 = matlabFunction(eq3);

eq4 = matlabFunction(eq4);

eq5 = matlabFunction(eq5);

% Задание сетки

l11=[-2:0.05:10];

l14=[-3:0.005:3];

[k11, k14] = meshgrid(l11,l14);

% Проверка устойчивости узлов сетки

c1 = eq1(k11,k14);

c2 = eq2(k11,k14);

c3 = eq3(k11,k14);

c4 = eq4(k11,k14);

c5 = eq5(k11,k14);

Kv11 = 1.6127;

Kv14 = -0.4477;

figure;

contourf(l11, l14, c1 & c2 & c3 & c4 & c5, [1 1 1 1 1])

colormap lines

hold on

pnt = scatter(Kv11, Kv14,'r','filled');

hold off

legend([pnt],"Выбранное значение коэффициентов")

xlabel("K11")

ylabel("K14")

Программа 7. Основной код для пункта 3

# **Приложение 4**

clc

clear

syms s z K11 K12 K14

% Зададим постоянные времени

Tn = 0.11;

Te = 0.11;

% Зададим время дискретизации

T0 = 0.18;

% Зададим матрицы системы уравнений

A = [-0.6, -5.71, -0.04, -2.9, 0.55;...

1, -0.26, 0.065, -0.04, 0;...

-0.7, -24, -2.5, -3.3, -19;...

0, 0, 0, -1/Tn, 0;...

0, 0, 0, 0, -1/Te];

B = [0, 0;...

0, 0;...

0, 0;...

1/Tn, 0;...

0, 1/Te];

C = [1, 0, 0, 0, 0;...

0, 1, 0, 0, 0;...

0, 0, 1, 0, 0;...

0, 0, 0, 1, 0;...

0, 0, 0, 0, 1];

D = 0;

I = eye(5, 5);

% Вычислим матрицу перехода G

G = 0;

n = 20;

for i = 0:n

G = G + A^i \* T0^i / factorial(i);

end

% Вычислим матрицу перехода Dd

Dd = 0;

for i = 0:n

Dd = Dd + A^(i) \* T0^(i+1) / factorial(i+1);

end

Dd = Dd \* B;

K11 = 1.6127;

K12 = 2.1617;

K14 = -0.4477;

K = [K11, K12, 0, K14, 0;

0, 0, 0, 0, 0];

G\_raz = G + Dd\*K;

W\_raz = collect((inv(z\*I-G\_raz))\*Dd);

Wx\_raz = W\_raz(3, 2);

tf\_Wx\_raz = convert\_to\_tf(Wx\_raz, true);

figure;

Wx\_raz\_w = (LAFCH(Wx\_raz, T0));

K23 = 0.05;

K = [K11, K12, 0, K14, 0;

0, 0, K23, 0, 0];

G\_zam = G + Dd\*K;

W\_zam = collect((inv(z\*I-G\_zam))\*Dd);

Wx\_zam = W\_zam(3, 2);

tf\_Wx\_zam = convert\_to\_tf(Wx\_zam, true);

Wx\_zam\_w = (LAFCH(Wx\_zam, T0));

figure;

step(d2c(tf\_Wx\_zam))

title('wx(t) при ступенчатом воздействии Ue');

grid on;

ylabel('wx(t)', Rotation=0);

Wy\_zam\_un = W\_zam(1, 1)

Wy\_zam\_ue = W\_zam(1, 2)

Wb\_zam\_un = W\_zam(2, 1)

Wb\_zam\_ue = W\_zam(2, 2)

Wx\_zam\_un = W\_zam(3, 1)

Wx\_zam\_ue = W\_zam(3, 2)

Wsn\_zam\_un = W\_zam(4, 1)

Wsn\_zam\_ue = W\_zam(4, 2)

Wse\_zam\_un = W\_zam(5, 1)

Wse\_zam\_ue = W\_zam(5, 2)

tf\_Wx\_zam\_un = convert\_to\_tf(Wx\_zam\_un, true)

tf\_Wx\_zam\_ue = convert\_to\_tf(Wx\_zam\_ue, true)

tf\_Wb\_zam\_un = convert\_to\_tf(Wb\_zam\_un, true)

tf\_Wb\_zam\_ue = convert\_to\_tf(Wb\_zam\_ue, true)

tf\_Wy\_zam\_un = convert\_to\_tf(Wy\_zam\_un, true)

tf\_Wy\_zam\_ue = convert\_to\_tf(Wy\_zam\_ue, true)

tf\_Wsn\_zam\_un = convert\_to\_tf(Wsn\_zam\_un, true)

tf\_Wsn\_zam\_ue = convert\_to\_tf(Wsn\_zam\_ue, true)

tf\_Wse\_zam\_un = convert\_to\_tf(Wse\_zam\_un, true)

tf\_Wse\_zam\_ue = convert\_to\_tf(Wse\_zam\_ue, true)

figure;

step(d2c(tf\_Wy\_zam\_un))

grid on;

title('wy(t) при ступенчатом воздействии Un');

ylabel('wy(t)', Rotation=0);

figure;

step(d2c(tf\_Wy\_zam\_ue))

grid on;

title('wy(t) при ступенчатом воздействии Ue');

ylabel('wy(t)', Rotation=0);

figure;

step(d2c(tf\_Wb\_zam\_un))

grid on;

title('b(t) при ступенчатом воздействии Un');

ylabel('b(t)', Rotation=0);

figure;

step(d2c(tf\_Wb\_zam\_ue))

grid on;

title('b(t) при ступенчатом воздействии Ue');

ylabel('b(t)', Rotation=0);

figure;

step(d2c(tf\_Wx\_zam\_un))

grid on;

title('wx(t) при ступенчатом воздействии Un');

ylabel('wx(t)', Rotation=0);

figure;

step(d2c(tf\_Wx\_zam\_ue))

grid on;

title('wx(t) при ступенчатом воздействии Ue');

ylabel('wx(t)', Rotation=0);

figure;

step(d2c(tf\_Wsn\_zam\_un))

grid on;

title('sn(t) при ступенчатом воздействии Un');

ylabel('sn(t)', Rotation=0);

figure;

step(d2c(tf\_Wsn\_zam\_ue))

grid on;

title('sn(t) при ступенчатом воздействии Ue');

ylabel('sn(t)', Rotation=0);

figure;

step(d2c(tf\_Wse\_zam\_un))

grid on;

title('se(t) при ступенчатом воздействии Un');

ylabel('se(t)', Rotation=0);

figure;

step(d2c(tf\_Wse\_zam\_ue))

grid on;

title('se(t) при ступенчатом воздействии Ue');

ylabel('se(t)', Rotation=0);

K2 = 2\*K;

G\_zam = G + Dd\*K2;

W\_zam = collect((inv(z\*I-G\_zam))\*Dd);

Wx\_zam = W\_zam(1, 2);

tf\_Wx\_zam = convert\_to\_tf(Wx\_zam, true);

Wx\_zam\_w = (LAFCH(Wx\_zam, T0));

figure;

step(d2c(tf\_Wx\_zam))

Wy\_zam\_un = W\_zam(1, 1)

Wy\_zam\_ue = W\_zam(1, 2)

Wb\_zam\_un = W\_zam(2, 1)

Wb\_zam\_ue = W\_zam(2, 2)

Wx\_zam\_un = W\_zam(3, 1)

Wx\_zam\_ue = W\_zam(3, 2)

Wsn\_zam\_un = W\_zam(4, 1)

Wsn\_zam\_ue = W\_zam(4, 2)

Wse\_zam\_un = W\_zam(5, 1)

Wse\_zam\_ue = W\_zam(5, 2)

tf\_Wx\_zam\_un = convert\_to\_tf(Wx\_zam\_un, true)

tf\_Wx\_zam\_ue = convert\_to\_tf(Wx\_zam\_ue, true)

tf\_Wb\_zam\_un = convert\_to\_tf(Wb\_zam\_un, true)

tf\_Wb\_zam\_ue = convert\_to\_tf(Wb\_zam\_ue, true)

tf\_Wy\_zam\_un = convert\_to\_tf(Wy\_zam\_un, true)

tf\_Wy\_zam\_ue = convert\_to\_tf(Wy\_zam\_ue, true)

tf\_Wsn\_zam\_un = convert\_to\_tf(Wsn\_zam\_un, true)

tf\_Wsn\_zam\_ue = convert\_to\_tf(Wsn\_zam\_ue, true)

tf\_Wse\_zam\_un = convert\_to\_tf(Wse\_zam\_un, true)

tf\_Wse\_zam\_ue = convert\_to\_tf(Wse\_zam\_ue, true)

figure;

step(d2c(tf\_Wy\_zam\_un))

grid on;

title('wy(t) при ступенчатом воздействии Un');

ylabel('wy(t)', Rotation=0);

figure;

step(d2c(tf\_Wy\_zam\_ue))

grid on;

title('wy(t) при ступенчатом воздействии Ue');

ylabel('wy(t)', Rotation=0);

figure;

step(d2c(tf\_Wb\_zam\_un))

grid on;

title('b(t) при ступенчатом воздействии Un');

ylabel('b(t)', Rotation=0);

figure;

step(d2c(tf\_Wb\_zam\_ue))

grid on;

title('b(t) при ступенчатом воздействии Ue');

ylabel('b(t)', Rotation=0);

figure;

step(d2c(tf\_Wx\_zam\_un))

grid on;

title('wx(t) при ступенчатом воздействии Un');

ylabel('wx(t)', Rotation=0);

figure;

step(d2c(tf\_Wx\_zam\_ue))

grid on;

title('wx(t) при ступенчатом воздействии Ue');

ylabel('wx(t)', Rotation=0);

figure;

step(d2c(tf\_Wsn\_zam\_un))

grid on;

title('sn(t) при ступенчатом воздействии Un');

ylabel('sn(t)', Rotation=0);

figure;

step(d2c(tf\_Wsn\_zam\_ue))

grid on;

title('sn(t) при ступенчатом воздействии Ue');

ylabel('sn(t)', Rotation=0);

figure;

step(d2c(tf\_Wse\_zam\_un))

grid on;

title('se(t) при ступенчатом воздействии Un');

ylabel('se(t)', Rotation=0);

figure;

step(d2c(tf\_Wse\_zam\_ue))

grid on;

title('se(t) при ступенчатом воздействии Ue');

ylabel('se(t)', Rotation=0);

Программа 8. Основной код для пунктов 4 и 6

function [outputArg] = LAFCH(W, T0)

syms w z lambda

z2w = (1 + w\*T0/2)/(1 - w\*T0/2);

H = subs(W, z, z2w);

[a, b] = numden(H);

c = coeffs(a, w);

c = max(c);

a\_H = vpa(collect(a / c));

b\_H = vpa(collect(b / c));

a\_H\_l = subs(a\_H, w, 1i\*lambda);

b\_H\_l = subs(b\_H, w, 1i\*lambda);

H = a\_H\_l / b\_H\_l;

F = matlabFunction(H);

x = logspace(-1,2,1000);

y = F(x);

L = 20\*log10(abs(y));

phi = [];

for i = 1:length(x)

if (real(y(i)) > 0 && imag(y(i)) > 0)

phi(i) = rad2deg(atan(imag(y(i))/real(y(i))));

end

if (real(y(i)) > 0 && imag(y(i)) < 0)

phi(i) = rad2deg(atan(imag(y(i))/real(y(i))));

end

if (real(y(i)) < 0 && imag(y(i)) > 0)

phi(i) = rad2deg(atan(imag(y(i))/real(y(i)))) + 180;

end

if (real(y(i)) < 0 && imag(y(i)) < 0)

phi(i) = rad2deg(atan(imag(y(i))/real(y(i)))) - 180;

end

end

figure;

subplot(2, 1, 1)

semilogx(x, L)

xlabel("lg(\lambda)")

ylabel("20 lg |W|, Дб", 'FontSize', 15)

grid on

subplot(2, 1, 2)

semilogx(x, phi)

xlabel("lg(\lambda)")

ylabel("\phi, градусы",'FontSize', 15)

ylim([-240 240])

grid on

outputArg = H;

end

Программа 9. Функция для построение псевдо-частотной логарифмической характеристики

# **Приложение 5**

clc

clear

syms s z K11 K12 K14

% Зададим постоянные времени

Tn = 0.11;

Te = 0.11;

% Зададим время дискретизации

T0 = 0.18;

% Зададим матрицы системы уравнений

A = [-0.6, -5.71, -0.04, -2.9, 0.55;...

1, -0.26, 0.065, -0.04, 0;...

-0.7, -24, -2.5, -3.3, -19;...

0, 0, 0, -1/Tn, 0;...

0, 0, 0, 0, -1/Te];

B = [0, 0;...

0, 0;...

0, 0;...

1/Tn, 0;...

0, 1/Te];

C = [1, 0, 0, 0, 0;...

0, 1, 0, 0, 0;...

0, 0, 1, 0, 0;...

0, 0, 0, 1, 0;...

0, 0, 0, 0, 1];

D = 0;

I = eye(5, 5);

% Вычислим матрицу перехода G

G = 0;

n = 20;

for i = 0:n

G = G + A^i \* T0^i / factorial(i);

end

% Вычислим матрицу перехода Dd

Dd = 0;

for i = 0:n

Dd = Dd + A^(i) \* T0^(i+1) / factorial(i+1);

end

Dd = Dd \* B;

K11 = 1.6127;

K12 = 2.1617;

K14 = -0.4477;

K23 = 0.05;

K = [K11, K12, 0, K14, 0;

0, 0, K23, 0, 0];

G\_zam = G + Dd\*K;

Kx = -0.2799;

W\_zam = collect((inv(z\*I-G\_zam))\*Dd);

Wx\_zam = W\_zam(3, 2);

tf\_Wx\_zam = convert\_to\_tf(Wx\_zam, true);

figure;

opt = stepDataOptions;

opt.StepAmplitude = Kx;

step(d2c(tf\_Wx\_zam), opt)

grid on;

title('wx(t) при ступенчатом воздействии Ue');

ylabel('wx(t)', Rotation=0);

Программа 10. Основной код для пункта 5