ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное автономное образовательное учреждение  
высшего профессионального образования**

**Национальный исследовательский университет  
«Высшая школа экономики»**

**Московский институт электроники и математики**

**Факультет Прикладной математики и кибернетики**

**Кафедра Кибернетики**

Дипломная работа

по специальности 230401.65 «Прикладная математика»

**Корректировка функционала качества системы с целью увеличения степени её устойчивости**

Студент группы М92 Елагин Н.Д.

Руководитель профессор, д.т.н.

Зотов М.Г.

Зав. кафедрой профессор, д.т.н.

Афанасьев В.Н.

**Аннотация**

на дипломную работу по специальности 230401.65 “Прикладная математика” студента группы М-92 Елагина Никиты Дмитриевича на тему:  
Корректировка функционала качества системы с целью увеличения степени её устойчивости (руководитель: проф., д.т.н. М.Г. Зотов). 2014.

Объектом исследования является класс управляемых объектов с полюсами на мнимой оси.

Цель работы – построение алгоритма составления функционала для решения задачи оптимальной фильтрации для указанного класса объектов как в пространстве состояний, так и в пространстве операторов, сравнительный анализ хода решения в обоих пространствах, а также поиск путей для построения стабилизирующего регулятора, обеспечивающего приемлемую степень устойчивости системы.

Теоретически разработанный алгоритм был подкреплён практическими примерами. Сконструированные системы были смоделированы и исследованы относительно вопроса абсолютной и относительной устойчивости.

Результатом работы также является разработанный в математическом пакете Matlab комплекс инструментов, позволяющий аналитически или численно строить фильтр Калмана и решать уравнения Винера-Хопфа.

Performance Index Adjustment for System's Stability Degree Increasing

by N.D. Elagin

**Abstract**

Aim of this work is to study control objects with poles on imaginary axes in the scope of building stabilizing regulator in signal optimal filtering problem. Performance indexes that can solve the task both in state and operator spaces are introduced. Their validity is proved with practice, during which constructed systems are modeled and approaches are compared relatively to stability attribute. A path to improve stability degree by adjusting weights in performance index is outlined. In addition a Matlab toolkit to build Kalman filter both analytically and numerically and to solve Weiner-Hopf equation is designed.

Supervisor: Prof. M.G. Zotov  
2014

№ страницы

Введение 6

Глава 1. Общие понятия 8

1.1 Линейная замкнутая система 8

1.1.1 Передаточная функция. 8

1.1.2 Характерестический полином 8

1.1.3 Управляемость 9

1.1.4 Наблюдаемость 9

1.2 Грубость системы 10

1.2.1 Определение. 10

1.2.2 Связь с критерием. 10

1.2.3 Устойчивость 11

1.3 Некоторые связи пространств операторов и состояний 12

1.3.1 Переход от дифференциального уравнения регулятора к передаточной функции 13

1.3.2 Переход от передаточной функции к дифференциальному уравнению в форме Коши 13

1.3.3 Реализация составляющих интегрального критерия в пространстве состояний 14

1.3.3.1 C использованием воздействий вида белых шумов 15

1.3.3.2 С использованием переменных состояния 16

1.4 Некоторые соотношения в задаче оптимальной фильтрации 17

1.4.1 Спектральная плотность случайных сигналов 17

1.4.2 Оценка дисперсии ошибки воспроизведения случайного сигнала. 18

Глава 2. Поиск оптимального регулятора в задаче фильтрации 19

2.1 Постановка задачи. 19

2.1.1 Пространство состояний 19

2.1.2 Пространство операторов 20

2.2 Построение алгоритма решения 21

2.2.1 Обзор классического подхода 21

2.2.2 Построение функционала качества 23

Глава 3. Разбор задачи в частном случае 29

3.1 Выбор программной среды 29

3.2 Разработка универсального способа решения 29

3.3 Анализ построенного алгоритма на примере 29

3.3.1 Условие 29

3.3.2 Первый способ: функционал без ограничений на компенсацию полюсов 30

3.3.3 Второй способ: функционал с ограничением на компенсацию полюсов, реализация через переменные состояний 37

3.3.4 Третий способ: функционал с ограничением на компенсацию полюсов, реализация с помощью воздействия вида белого шума 45

Заключение. 49

Список литературы. 50

Приложение. 51

**Введение**

Теория автоматического управления – молодая наука, оформившаяся в самостоятельную общенаучную дисциплину в начале XX века. Истоки её инженерных методов лежат в Общей теории линейных регуляторов, представленной работами Д. Максвелла и И. Вышеградского. Она интенсивно развивается и по сей день, в зависимости от объекта исследований изменяя взгляды как на сам предмет, так и на математический аппарат. В конце 1950-х годов в связи с развитием ракетной промышленности и космонавтики зародился новый аппарат описания системы управление – управление в пространстве состояний. Именно в это время возникает фундаментальная идея оптимальности, как оптимизация некоторого качества выбором управления. Проблема созданной теории заключалась в том, что она позволяла адекватно решать лишь узкий круг практических задач, а использованию стройной математической теории оптимального управления препятствовали неизбежная неопределённость измерений как параметров самой модели, так и наблюдаемых состояний/выходов. Появилась задача синтеза как необходимость построения управления в форме обратной связи, т.е. функции от выхода системы в текущий момент.

Дальнейшее развитие приводит к созданию -теории, объединяющей частотные методы и методы пространства состояний, что позволило рассматривать различные задачи с неопределённостью [5].

Одной из основных задач управления линейными объектами является задача о стабилизации, которая напрямую связана с важной - если не первостепенной - характеристикой при анализе и синтезе систем управления с обратной связью - устойчивостью. Существует большой класс современных реальных систем, таких как, например, истребители, которые без использования активной обратной связи являются неустойчивыми [1]. Поэтому вопросы как критики, так и совершенствования подходов к удовлетворению системой свойства устойчивости являются крайне актуальными. Замкнутая система всегда может быть однозначно определена как (абсолютно) устойчивая или неустойчивая. В случае установленной абсолютной устойчивости может идти речь о её степени – об относительной устойчивости. В простейшем определении динамическая система полагается устойчивой, если она обладает ограниченной реакцией на ограниченное воздействие.

Для конструирования устойчивой системы требуется базовый функционал качества, который в дальнейшем будет подвергаться корректировке. Одним из наиболее простейших и изученных является квадратичный функционал, исследованный в работах Калмана, Беллмана, Летова, Виллемса. Задачу конструирования линейно-квадратично регулятора можно решить как в пространстве состояний, так и в пространстве операторов. В первом случае решение связано с уравнением Риккати, во втором – с уравнением Винера-Хопфа.

С развитием двух подходов возникает необходимость их сравнительной оценки, поэтому настоящая работа ставит себе целью рассмотреть проблематику конструирования оптимального регулятора для объекта управления при условии придания системе свойства грубости и обеспечения приемлемой степени устойчивости. Решение в пространстве изображений будет искаться в множестве управляющих устройств с недостаточным числом звеньев коррекции в предположении, что корни объекта управления могут находится в левой полуплоскости и на мнимой оси.

**Глава 1. Общие понятия**

* 1. **Линейная замкнутая система**

Рассмотрим линейную стационарную непрерывную управляемую систему:

|  |  |
| --- | --- |
|  | (1.1) |

– вектор состояния системы, – управление, – выход системы, – входные сигналы (внешние возмущения) или задающие воздействия.- ектор состт времени к Матрицы не зависят от времени t. Подобную запись назовём описанием в пространстве состояний. Под операторным описанием подразумевается запись соотношения вход-выход посредством линейного оператора [5]:

* + 1. **Передаточная функция**

Определим оператор дифференцирования по времени и положим s комплексной переменной. Тогда матричная функция

называется передаточной функцией от управления к выходу [5]. Аналогично

называется передаточной функцией от возмущения к выходу [5].   
 – матрица, элементами которой являются дробно-рациональные функции. С помощью передаточных функций можно записать выход системы как функцию от управления и внешних входов:

* + 1. **Характеристический полином**

Элементами матриц являются дробно-рациональные функции от переменной s, которые имеют общий знаменатель

Его называют характеристическим полиномом системы (1.1) [5].

* + 1. **Управляемость**

Проблема управляемости – это ликвидация начального рассогласования [4]. Рассматривается это понятие через программное управление (поиск управления как функции от времени). Рассмотрим систему

|  |  |
| --- | --- |
|  | (1.2) |

Назовём её управляемой, если начальное отклонение может быть устранено за любое конечное время.

**Теорема (критерий управляемости)** [5]

*Система (1.2) управляема тогда и только тогда, когда пара (A, B) невырождена:*В операторном описание условие управляемости можно представить следующим образом: [1].

Это условие удобно в силу того, что сведение операторного описания к пространству состояний не всегда единственно, а вышеозначенный критерий позволяет не вычислять форму Мак-Миллана для установления минимальной реализации.

Данные выше критерий описывает условие управляемости по состояниям. В случае системы (1.1), когда управление строится по выходу с помощью, например, фильтра Калмана, важно условие управляемости по выходу, которое даётся следующим образом для систем с -мерным выходом [4]:

* + 1. **Наблюдаемость**

Пусть недоступно (или слишком дорогостояще) измерение состояния системы (1.2). И единственная предоставляемая информация – её выход:

|  |  |
| --- | --- |
|  | (1.3) |

Классическое определение – если разрешима задача вычисления по точным измерения выхода , входа и их производных, то система (1.3) наблюдаема.

**Теорема (критерий наблюдаемости)** [5]

*Система (1.3) наблюдаема тогда и только тогда, когда пара невырождена:*В операторном описание условие управляемости можно представить следующим образом: [1].

Важный смысл совокупности свойств управляемости и наблюдаемости – возможность разместить все корни характеристического полинома в заданных точках s-плоскости [1]. Что напрямую влияет на возможность придания данной системе устойчивости, т.к. она зависит от расположения корней полинома.

* 1. **Грубость системы**

Ввиду неточности определений характеристик управляемого объекта, неполноты его описания, наличия возмущений различного рода, деградаций самой элементной базы, на которой было реализовано управление [2], а также ограниченной точности вычислений характеристик регуляторов ввиду не всегда возможного аналитического решения и, как следствие, применения численных методов для построения управляющего устройства, реальные – действительные – параметры системы будут отклоняться от модели. Ввиду этого особое значение приобретает понятие грубости. Дадим его определение:

* + 1. **Определение**

Пусть система характеризуется некоторым набором параметров и пусть она обладает свойством для некоторого , тогда система называется *грубой* относительно свойства , если оно выполняется во всех точках некоторой окрестности точки [2].

* + 1. **Связь с критерием качества**

Понятие грубости тесно связано с понятием чувствительности, в том числе чувствительности показателей качества к изменению параметров системы.

Показатель качества системы называется грубым, если он непрерывно зависит от изменения динамических характеристик её составных частей. В технической постановке конструирования грубость показателя может оцениваться допустимым изменением его значения при реальных изменениях динамических параметров системы. Необходимым условием грубости показателя качества системы является грубость системы относительно свойства устойчивости [2].

* + 1. **Устойчивость**

Перейдём от общих понятий к формализации данного свойства в интересующей нас сфере. В случае нестационарных систем существует множество различных определений устойчивости, например, устойчивость по Ляпунову и асимптотическая устойчивости. В рассматриваем случае линейных стационарных систем все определения совпадают и речь ведётся просто об устойчивости [5].

В данной работе свойство устойчивости рассматривается в задаче конструирования системы с обратной связью при наличии возмущений для воспроизведения задающего воздействие, т.е. проектируется датчик. Естественно, подобная обратная связь также должна быть стабилизирующей, т.е. придающей системе свойство абсолютной устойчивости. Приведём определение устойчивости [4]:

Будем полагать систему (1.1) *устойчивой*, если она

1. устойчива по начальным условия, т.е. при имеет место
2. устойчива по входу, т.е. из ограниченности воздействия следует ограниченность всех фазовых переменных и компонент выхода

Рассмотрим теперь вопрос устойчивости в пространстве операторов. Описание в нём связано с понятиями “вход-выход”. Его суть сводится к представлению системы в виде звеньев: , где – комплексная передаточная функция звена, – выход системы, – задающее воздействие.

Звено называется *устойчивым по входу* [4], если при любом ограниченном входном воздействии и нулевых начальных условиях выходная реакция является ограниченной при любом конечном и при   
.

**Критерий устойчивости по входу** [4]

*Устойчивость по входу имеет место, если выполнены два условия:*

1. *степень полинома в знаменателе должна быть не меньше степени полинома в её числителе (условие строгой реализуемости звена).*
2. *Все корни характеристический полинома звена лежат в левой полуплоскости (условие устойчивости характеристичского полинома).*

**Критерий устойчивости по начальным условиям** [4]

*Устойчивость по начальным условиям имеет место, если характеристический полином устойчив.*

В настоящей работы выполнение условия грубости системы относительно свойства устойчивость будет проверяться непосредственным отысканием корней характеристического полинома всей системы.

Рассмотрим также понятие *относительной устойчивости*, которое характеризует степень устойчивость системы. Данное свойство имеет место только при наличии абсолютной устойчивости и оценивается численно как величина действительной части ближайшего к мнимой оси корня характеристического полинома системы. Физический смысл – запас грубости системы относительно свойства устойчивости.

* 1. **Некоторые связи пространств операторов и состояний**

При формулировании задачи в одном из пространств возникает необходимость перехода к тождественной постановки в другом. К сожалению, не всегда исчерпывающее описание из, например, пространства изображений, будет таким же в пространстве оригиналов. Яркий пример – ограничение на управление, которое в случае решения задачи в терминах вход-выход можно опустить, если известно из условий, что оно конечно. Классический же алгоритм построения наблюдателя и управления в пространстве состояний требует наличия данного ограничения.

Более того, проверку системы на устойчивость весьма удобно производить в пространстве операторов ввиду наглядности получения характеристического полинома и его анализа. Поэтому необходимо уметь представлять построенный в пространстве состояний регулятор в терминах вход-выход. Данное представление в виде передаточных функций позволяет легко сравнивать решения, полученные в обоих пространствах.

* + 1. **Переход от дифференциального уравнения регулятора к передаточной функции**

Рассмотрим общий вид оптимального регулятора системы (1.1) при (т.е. воздействие – скаляры):

|  |  |
| --- | --- |
|  | (1.4) |

где , – наблюдаемый выход, – матрицы, полученные при решении уравнений Рикатти соответствующей оптимизационной задачи с критерием .

Проведём преобразование Лапласа всех уравнений (1.4):

|  |  |
| --- | --- |
|  | (1.4.1)  (1.4.2) |

Мы получили систему из линейных относительно уравнений. Из (1.4.1) выражаем через и полученный результат подставляем в (1.4.2). В итоге найдём: . – и будет искомой передаточной функцией регулятора. Заметим, что подобным образом можно заменить систему наблюдателя на одно звено и уменьшить сложность управляющего устройства и, как следствие, ошибки и запаздывание управления [2].

* + 1. **Переход от передаточной функции к дифференциальному уравнению в форме Коши**

Рассмотрим произвольное звено коррекции:

|  |  |
| --- | --- |
|  | (1.5) |

Соотношение между и обусловлено требованием строгой реализуемости проектируемого звена как необходимого условия для устойчивости конструируемой системы.

Переход к дифференциальному уравнению в форме Коши производится на основе алгоритма оценки сложности управляющего устройства на элементах аналоговой техники, построенной на операционных усилителях.

Для этого представим (1.5) в виде дифференциального уравнения:

Разрешим это уравнение относительно старшей производной :

Используя полученное выражение, можно построить соответствующую схему с использование интеграторов, сумматоров и умножителей. Реализацию дифференциального уравнения на дискретной технике можно представить в форме Коши [2]:

* + 1. **Реализация составляющих интегрального критерия в пространстве состояний**

В случае, когда исходная постановка задачи даётся в пространстве операторов, удобно составлять интегральный критерий относительно желаемых свойств системы, описанных в терминах вход-выход, а потом переводить постановку задачи (как совокупность описания системы и критерия) в пространство состояний для сравнительного решения. Это оправдывается тем, что операторный подход позволяет весьма гибко и – самое главное – наглядно строить функционал качества. Более того, рассматриваемая проблематика компенсации регулятором находящихся на мнимой оси нулей и полюсов управляемого объекта имеет известное решение. К сожалению, не все линейные комбинации частных критериев могут быть реализованы, а те из них, реализация которых возможна, требуют некоторой доли изобретательности [2]. Рассмотрим несколько подобных приёмов.

* + - 1. **C использованием воздействий вида белых шумов**

Приведём некоторые оценки, линейные комбинации которых используются в качестве обобщённого интегрального критерия при решении задач в пространстве операторов, и их смысл в пространстве состояний:

Представляет собой дисперсию разности выходных координат желаемой и действительной систем при подаче на их входы одного и того же белого шума с единичной интенсивностью [2].

Представляет собой дисперсию *i*-ой производной сигнала управления при подаче на выход датчика белого шума единичной интенсивности. Датчик замеряет выходной сигнал объекта управления [2].

|  |  |
| --- | --- |
|  | (1.6) |

Представляет собой дисперсию сигнала на выходе цепочки последовательно соединённых интегрирующих устройств, на вход которой подаётся выходная координата системы. На вход объекта управления в системе подаётся белый шум единичной интенсивности [2].

Благодаря подобным связям появляется возможность либо непосредственного перевода оценок на язык пространства состояний – в том случае, если в системе уже постановкой условия заданы нужные воздействия, либо с помощью добавления фиктивного воздействия, которое приведёт к пополнению функционала нужными оценками.

* + - 1. **С использованием переменных состояния**

Метод добавления фиктивного воздействия предоставляет возможность замерять различные параметры системы. Возникает вопрос как учесть в пространстве состояний реакцию системы на подобное добавление. Оказывается, в некоторых случаях, например для оценки (1.6), это можно сделать с помощью переменных состояния [2].

Рассмотрим дифференциальное уравнение порядка :

|  |  |
| --- | --- |
|  | (1.6) |

Преобразуем его по Лапласу:

Его решение в пространстве изображений имеет вид:

А в пространстве оригиналов запишется так:

Из приведённых выше выкладок видно, что воздействие на систему вида

Может быть получено за счёт расширения пространства состояний. Начальные условия задаются соотношениями .

Приведём теперь вид уравнения (1.7) в пространстве состояний:

* 1. **Некоторые соотношения в задаче оптимальной фильтрации**

Задача оптимальной фильтрации заключается в конструировании устройства, отфильтровывающий полезный сигнал от помехи. Приведём схему для множества структур управления с недостаточным числом звеньев коррекции:

Рисунок 1

Воздействие , – сигналы, получаемые из белых шумов. – ошибка воспроизведение, – желаемый полезный сигнал.

Заметим, что, например, если положить , то решение соответствующей задачи будет обеспечивать минимум ошибки воспроизведения задающего воздействия. Если же , то мы приходим к задаче воспроизведения производной полезного сигнала

* + 1. **Спектральная плотность случайных сигналов**

Пусть – стационарный центрированный случайный процесс. – его реализация. Введём также его усечённую реализацию:

и – результат преобразования Фурье от него. Тогда спектральную плотность сигнала можно найти как:

В случае, если ввести в рассмотрение ещё один случайный процесс , то их взаимную спектральную плотность можно найти следующим образом:

* + 1. **Оценка дисперсии ошибки воспроизведения случайного сигнала**

Рассмотрим сигнал . Он является результатом прохождения сигналов воздействия через нашу систему. Зная его спектральную плотность, можно оценить его дисперсию:

|  |  |
| --- | --- |
|  | (1.7) |

На основании схемы (рис. 1) найдём :

|  |  |
| --- | --- |
|  | (1.8) |

Подставляя (1.8) в (1.7) получим:

|  |  |
| --- | --- |
|  | (1.9) |

**Глава 2. Поиск оптимального регулятора в задаче фильтрации**

Сформулируем задачу поиска оптимального регулятора в общих понятиях: дан многомерный реальный объект управления с квадратной матричной передаточной функцией . Необходимо, имея математическую модель объекта, т.е. передаточную функцию найти расчётные значения параметров передаточных функций входящих в управляющее устройство звеньев коррекции так, чтобы построенное по найденной математической модели реальное управляющее устройство обеспечивало в реальной системе управления (реальный объект с реальными управляющим устройством) характеристики, близкие к расчётным [2]. Стоит уточнить, что речь идёт о конструировании управляющего устройства как элемента возможного множества. В настоящей работе рассматривается проблематика конструирования датчика, замеряющего входное воздействие. Решение (управляющее звено) будет искаться как элемент множества управляющих устройств с недостаточным числом звеньев коррекции. Включаться оно будут в прямую цепь.

* 1. **Постановка задачи**

Изложим постановки интересующей нас задачи в обоих пространствах. Предварительно приведём классическую схему многомерной системы управления.

Рисунок 2: Классическая схема многомерной системы управления

* + 1. **Пространство операторов**

На ней – квадратная матрица передаточных функций объекта управления с полюсами и нулями на мнимой оси. – матрица передаточных функций фильтра, формирующего вектор случайного воздействия из случайного стационарного процесса вида белого шума. - матрица передаточных функций фильтра, формируеющего из случайного стационарного процесса вида белого шума вектор помехи , наложенной на управляющее воздействие . – управление; – выход; наблюдаемый вход в регулятор. – передаточная функция регулятора.

Необходимо найти такую , которая бы обеспечивала минимизацию ошибки воспроизведения задающего воздействия и придавала бы системе свойство устойчивости.

* + 1. **Пространство состояний**

Постановка в пространстве состояний даётся через объект с расширением, включающим в себя помимо системы дифференциальных уравнений, соответствующих передаточной функции объекта , описание формирующих фильтров :

|  |  |
| --- | --- |
| , | (2.1) |

– вид доступного измерения,  *–* вектор линейной комбинации случайных стационарных процессов типа белого шума и таких, что

|  |  |
| --- | --- |
|  | (2.2) |

Где – операция математического ожидания.

Необходимо построить наблюдателя вида

|  |  |
| --- | --- |
|  | (2.3) |

Обеспечивающего устойчивость системы и минимум ошибки воспроизведения задающего воздействия.

* 1. **Построение алгоритма решения**
     1. **Обзор классического подхода**

Приведём теорему для формирования линейного закона управления с обратной связью в пространстве состояний [3]:

*Дан объект, описываемый (2.1), (2.2). Пусть пара матриц управляема, наблюдаема, невырожденна, а матрица интенсивностей положительно определённая. Тогда закон управления (2.3) доставляет минимум показателю*

|  |  |
| --- | --- |
|  | (2.4) |

*Матрица коэффициентов усиления в обратной связи K находится из соотношений*

|  |  |
| --- | --- |
|  | (2.5) |

*А матрица коэффициентов в устройстве оценивания находится из*

|  |  |
| --- | --- |
|  | (2.6) |

*– положительно определённые решения соответствующих уравнений.*

Рассмотрим классический подход к решению поставленной в 2.1.2 задаче. Требуется обеспечить минимум ошибки воспроизведения задающего воздействия. Пусть описание (2.1) уже является расширенным, т.е. учитывает формирующие фильтры. Пусть , – координата вектора отвечает за задающее воздействие, – наблюдаемый выход. Тогда для решения задачи можно использовать следующий функционал:

|  |  |
| --- | --- |
|  | (2.7) |

Но, как показывает практика (в Главе 3 будут приведены примеры), если у матрицы из (2.1) есть собственные значения на мнимой оси, то полученный с помощью критерия качества (2.7) линейный регулятор не будет обладать стабилизирующими качествами.

Перейдём теперь к рассмотрению классического подхода при решении задачи в пространстве операторов. Суть его заключается в отыскании матричной передаточной функции , , обеспечивающий минимум ошибки воспроизведения на выходе системы задающего воздействия . По найденной и известной определяется искомая матрица . На основе риc. 1 найдём . Для этого запишем уравнения относительно входа-выхода:

Отсюда

Или для одномерного случая:

|  |  |
| --- | --- |
|  | (2.8) |

Передаточной функции соответствует в пространстве состояний система наблюдателя (2.3).

Т.к. задача конструируемого датчика – отфильтровать полезный сигнала от помехи , то нам надо оценить дисперсию ошибки воспроизведения сигнала и построить на данной оценке функционал.

Для этого обратимся к (1.9). Для использования этой оценки в функционале нам необходимо найти .

С учётом постановки задачи:

и того, что сигналы полагаются некоррелированными, получим:

где . – спектральная матрица векторного сигнала .

Тогда сам функционал запишется как:

|  |  |
| --- | --- |
|  | (2.9) |

Здесь первая составляющая функционала оценивает качество слежения, вторая – мощность управления.

При построении по критерию (2.9) регулятора для объекта управления с полюсами на мнимой оси было выявлено (глава 3), что сконструированная система оказывается неустойчивой – передаточная функция компенсировала полюса объекта управление. Как следствие характеристический полином систем оказался неустойчивым.

Напоследок заметим, что критерий (2.7) адекватен (2.9) в том смысле, что полученные с помощью них регуляторы будут эквивалентны.

* + 1. **Конструирование функционала**

Основная часть алгоритма – не связанная с математическими сложностями вычисления решений уравнений Риккати и Винера-Хопфа – заключается в составлении правильного функционала и решении задачи с его помощью.

Так как было установлено, что функционалы (2.7) и (2.9) не позволяют решить поставленную задачу, возникает вопрос о там как именно стоит их модифицировать, чтобы сконструированный регулятор обладал стабилизирующим свойством.

Для того, чтобы разрешить проблему, мы проанализируем свойства, которыми должен быть обладать регулятор, дополним функционал в операторной постановке задачи и приведём способ его перевода в постановку пространства состояний. Используя полученные выводы можно будет провести сравнительную оценку практического хода решения задачи в обоих пространствах, а также проверить составленные функционалы на адекватность относительно желаемых свойств сконструированной системы (глава 3).

Вернёмся к постановке задачи. Нам требуется построить линейный квадратичный регулятор, который бы:

1. обеспечивал минимум воспроизведения задающего воздействия, т.е. минимизировал ,
2. обладал стабилизирующим свойством.

Первое условие уже учитывается в базовом функционале в обоих постановках. Рассмотрим техническую сторону второго требования. Стабилизирующий регулятор должен обеспечивать свойство устойчивости в системе. Систему мы считаем устойчивой, если она обладает этим свойством по начальным условия и по входу. Заметим, что – на основании критериев – устойчивость по начальным условиям следует из устойчивости системы по входу – так как указанное имеет место при устойчивости характеристического полинома системы. Таким образом регулятор должен обеспечить две вещи:

1. устойчивость характеристического полинома системы,
2. свою строгую реализуемость.

Строгая реализуемость регулятора обеспечивается второй составляющей критерия (2.9). Чтобы необходимо, чтобы степень полиномов в числителе элементов матрицы как минимум не превосходила степени полиномов в знаменателе.

Для устойчивости характеристического полинома системы необходимо и достаточно, чтобы все его корни имели отрицательную действительную часть. В случае, если построенный регулятор компенсирует находящиеся на мнимой оси нули или полюса объекта управления, полином становится неустойчивым, так как у него появляются корни, не удовлетворяющие критерию.

Рассмотрим одномерный случай. Запишем характеристического уравнение нашей замкнутой системы:

Тогда, полагая , запишем уравнение для поиска корней характеристического полинома:

Пусть – находящиеся на мнимой оси и компенсируемые построенным регулятором нули и полюса объекта управления, тогда:

Отсюда

Как мы видим, если регулятор компенсирует находящиеся на мнимой оси нули или полюса управляющего объекта, они автоматически станут корнями характеристического полинома системы и, как результат, мы получим неустойчивую систему.

Таким образом нам необходимо дополнить функционал (2.9) таким образом, чтобы он исключал возможность возникновения рассмотренной выше ситуации.

Заметим, что критерий (2.9) уже ограничивает возможность компенсации нулей управляющего устройства. Вторая его составляющая

конечна, если не имеет полюсов на мнимой оси. Таким образом ограничение на компенсацию регулятором правых нулей объекта управления уже учтено как в пространстве операторов, так и в пространстве состояний. Проверим это утверждение для одномерного случая:

Из условия сходимости оценки должен компенсировать лежащие на мнимой оси нули полинома . Как итог – они не попадут в знаменатель .

Итак, было установлено, что функционал необходимо пополнить условием, которое бы ограничивало компенсацию регулятором мнимых полюсов устройства управления.

Для избавления от возникающей негрубости системы можно ввести в функционал (2.9) с малым весом оценку для следующего выражения:

|  |  |
| --- | --- |
|  | (2.11) |

Для ограниченности её оценки, т.е.

|  |  |
| --- | --- |
|  | (2.12) |

Необходимо, чтобы полюса выражения (2.11) не лежали на мнимой оси. Обратимся к одномерному случаю:

Сформулированное выше условие выполняется, если компенсирует мнимые нули . И на основе выражения для из (2.8) можно утверждать, что введение оценки (2.12) в функционал (2.9) обеспечит ограничение на компенсацию регулятором мнимых полюсов устройства управления.

Приведём модифицированный функционал:

|  |  |
| --- | --- |
|  | (2.13) |

Реализовать в пространстве состояний оценку (2.12) можно с помощью переменных состояния – расширением пространства, либо с помощью воздействия вида белого шума.

Чтобы перевести добавку (2.12) на язык пространства состояний первым способом необходимо будет ввести набор переменных, которые будут реализовывать (на основе уже имеющегося вектора состояний)  
. В итоге у нас будет переменная , такая что:

Более подробно данный вопрос будет рассмотрен в главе 3.

К сожалению, было установлено, что перевод функционала (2.13) в пространстве изображений с помощью расширения вектора состояний не дал желаемого результата – полученный регулятор не обладал стабилизирующим свойством.

Обратимся ко второму способу – использованию воздействия вида белого шума. Некоторые составляющие интегрального критерия могут быть реализованы в пространстве состояний с помощью введения фиктивного воздействия. Рассмотрим (1.6). Для её реализации в постановке нашей задачи необходимо подать на вход объекта белый шум единичной интенсивности.

Приведём схему подобного устойства:

Рисунок 3

Подобное фиктивное воздействие приведёт к тому, что функционал примет вид:

|  |  |
| --- | --- |
|  | (2.14) |

Последняя составляющая обусловлена вкладом вводимой помехи в сигнал управления .

Приведём вывод этой оценки. Для этого найдём передаточную функцию от входа к выходу :

Теперь выразим :

Теперь вклад сигнала в управление можно оценить:

**Глава 3. Разбор задачи в частном случае**

В настоящей главе будет произведён разбор частного случая задачи оптимальной фильтрации. На примере будет разобран ход построения алгоритма, будут получены и практические результаты, на которые мы ссылались в предыдущей главе и на которые опирались при конструировании функционала качества для решения поставленной задачи. Математические выкладки и результаты будут также подкреплены компьютерным моделированием сконструированных систем.

* 1. **Выбор программной среды**

В качестве вспомогательной программной среды для решения задачи был выбран пакет Matlab. Выбор обоснован тем, что это мощный инструмент, позволяющий разрешать поставленные проблемы не только аналитически, но и численно. Обширная библиотека как точных, так и численных методов позволяет вычислять нужные результаты с приемлемой точностью. Также Matlab содержит уже готовый функционал для построения линейного квадратичного регулятора в случае пространства состояний. Более того с помощью входящего в состав Matlab инструмента Simulink мы проведём моделирование сконструированных систем управления и проверим интересующие нас свойства.

* 1. **Разработка универсального способа решения**

Для ускорения процесса конструирования регулятора в пространстве состояний в Matlab была разработана функция, которая, при должной настройке, позволяет разрешать поставленную проблему как аналитически, так и численно. Также был реализован функционал, позволяющий решать уравнение Винера-Хопфа.

* 1. **Анализ построенного алгоритма на примере**
     1. **Условие**

Пусть передаточная функция объекта управления задана:

Задающее воздействие случайное с нулевым математическим ожиданием. Формируется из белого шума единичной интенсивности. Имеет спектральную плотность

На задающее воздействие наложен белый шум единичной интенсивности. Шумы – некоррелированны. Необходимо построить оптимальный стабилизирующий регулятор, обеспечивающий в системе минимум ошибки воспроизведения задающего воздействия . Решение ищется в множестве управляющих устройств с недостаточным числом звеньев коррекции.

Приведём структурную схему конструируемой системы:

Рисунок 4: Схема конструируемой системы

* + 1. **Первый способ: функционал без ограничений на компенсацию полюсов.**

Попробуем решить поставленную задачу, используя в качестве критерия качества функционал (2.9).

*Пространство состояний.*

Переведём условия на язык пространства изображений. Начнём с передаточной функцию объекта. Для этого воспользуемся алгоритмом, приведённым в пункте 1.3.2:

Воспользуемся записью в форме Коши:

|  |  |
| --- | --- |
|  | (3.1) |

Теперь расширим описание воздействием. Сигнал проходит через формирующий фильтр и приобретает спектральную плотность . Тогда:

И система запишется:

Осталось определить наблюдаемый выход. Для этого используем нерасширенное представление (3.1) и найдём матрицу передаточных функций:

Отсюда и согласно рис. 4 доступно измерение вида:

Также, согласно условию:

Перейдём теперь непосредственно к решению. Установим свойства системы:

Система наблюдаема.

Для управляемости системы, в случае, если информация ограничена выходом системы, т.е. наблюдаемым , необходимо и достаточно, чтобы тройка была невырожденна.

Система управляема.

Проверим последнее условие:

Пара матриц оказалась вырожденной. Тем не менее продолжим решение и покажем, что построенный регулятор не будет обладать стабилизирующим свойством.

Критерий качества из (2.7) запишется:

Отсюда.

Для построение регулятора воспользуемся функционалом Matlab – функцияими *lqe* и *lqr.*

С помощью процедуры *lqr* был найден вектор:

Попытка найти коэффициенты обратной связи с помощью процедуры *lqe* не увенчалась успехом. Поэтому было решено вручную найти с помощью соотношений (2.6). Проблема была разрешена численно с помощью функции build\_regulator (см. Приложение А Листинг 1).

Теперь закон управления согласно (2.3) запишется в виде:

Воспользуемся теперь алгоритмом, изложенным в пункте 1.3.1 для того, чтобы найти передаточную функцию, эквивалентную регулятору.

Составим систему линейных уравнений:

Разрешив её относительно , и с учётом того, что получим:

Проверим данный регулятор на устойчивость по входу. Условие строгой реализуемости выполняется. Осталось найти корни характеристического полинома:

Все корни лежат в левой полуплоскости – можно утверждать, что звено устойчиво по входу.

Найдём теперь корни числителя:

Как мы видим, первые два корня с точностью в по модулю совпадают с полюсами объекта управления . Данная неточность является следствием решения проблемы численным методов.

Появления полюсов объекта управления в числителе передаточной функции построенного регулятора было предсказано нами ранее. Сконструированная подобным образом (т.е. на основе критерия качества (2.7)) не обладает свойством устойчивости.

*Пространство операторов*

Разрешим теперь проблему в пространстве изображений, используя функционал (2.9):

Запишем уравнение Винера-Хопфа:

Отсюда

И

В итоге получим:

Найдём корни числителя:

Как мы видим, найденный регулятор компенсирует полюса объекта управления.

Таким образом было установлено, что функционал вида (2.9) не позволяет решить поставленную задачу.

Заметим отличия регулятора, полученного при решении задачи в пространстве оригиналов и пространстве изображений. Ввиду вынужденного (аналитически задачу разрешить не удалось) численного решения системы нелинейных уравнений для поиска коэффициентов построенное управление потеряло в точности и, как следствие, качестве. Хотя для эквивалентной по постановке задаче в пространстве операторов нашлось точно решение, обеспечивающие приемлемую аккуратность слежения. Приведём также построенную в Simulink схему и графики смоделированных процессов:

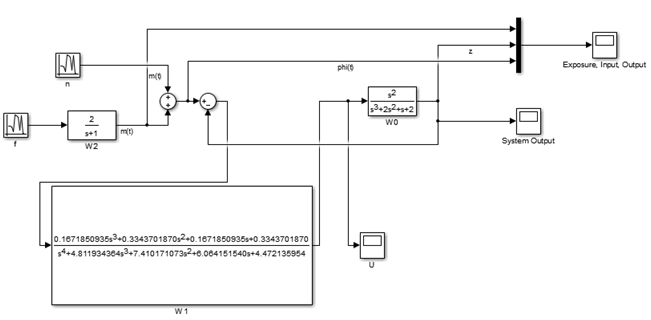


Рисунок 5: Модель сконструированной системы в пространстве операторов

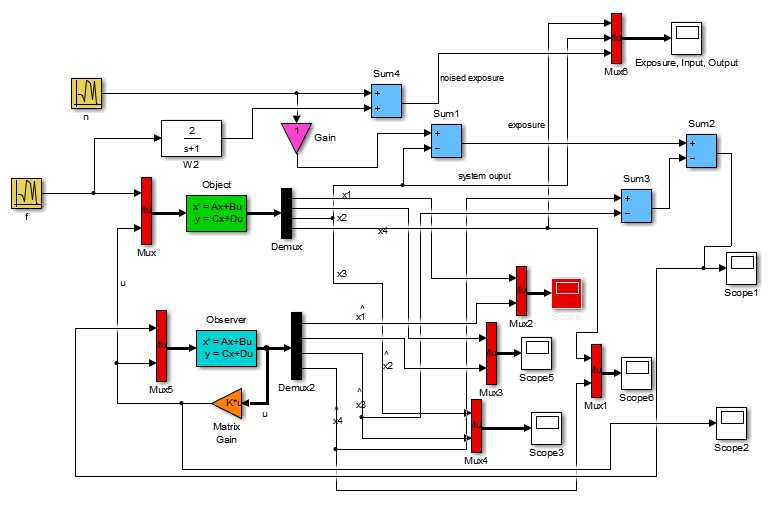


Рисунок 6: Модель сконструированной системы в пространстве состояний

Представим полученный при моделировании график, выводимый элементом “Exposure, Input, Output”:

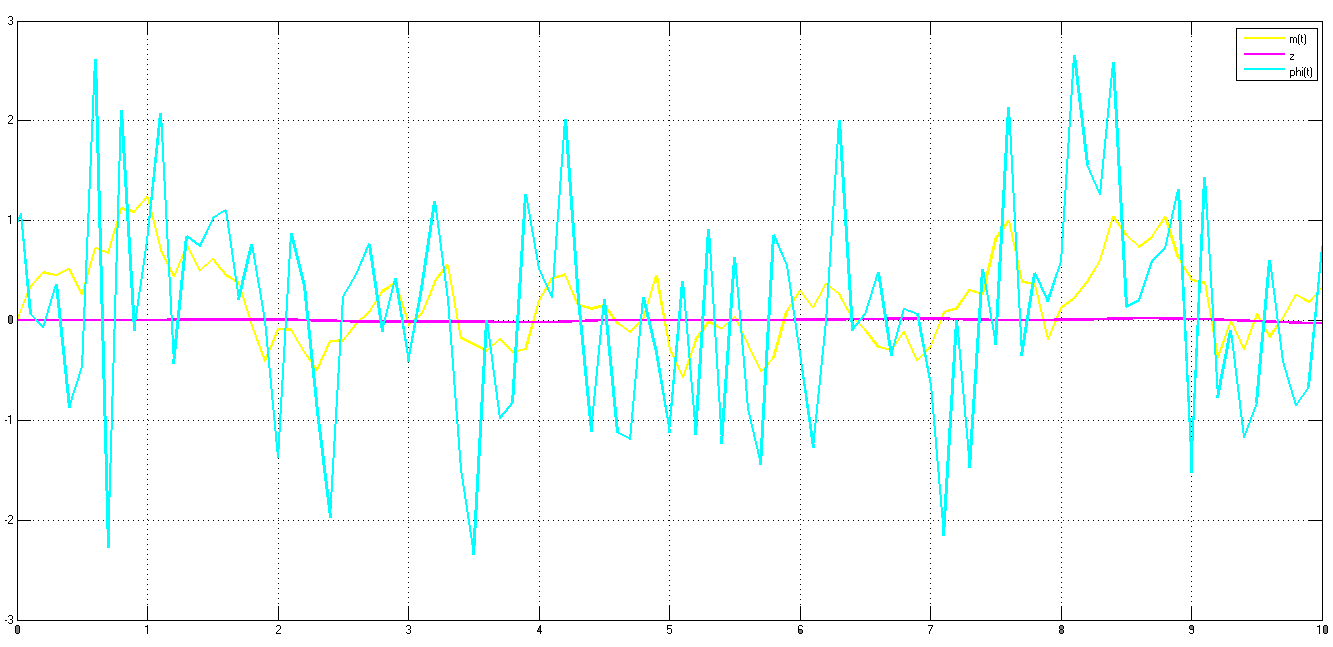


Рисунок 7: График слежения №1 (пространство изображений)

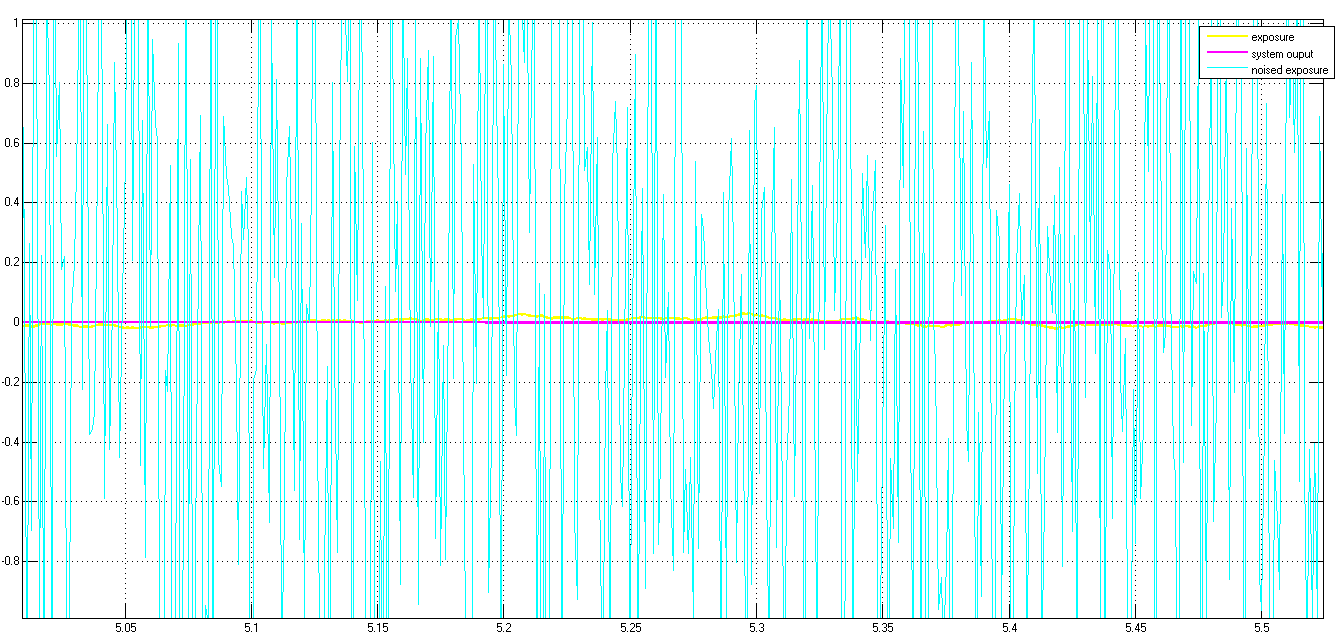


Рисунок 8: График слежения №2 (пространство оригиналов)

На рис. 7-8 на оси абсцисс отложено время, на оси ординат – значения наблюдаемых величин. График бирюзового цвета, задающего воздействия – жёлтый, выход системы , воспроизводящий задающее воздействие, отмечен сиреневым.

* + 1. **Второй способ: функционал с ограничением на компенсацию полюсов, реализация через переменные состояний.**

Воспользуемся теперь для поиска решения функционалом (2.13).

*Пространство операторов*

В рассматриваем примере добавка (2.12) может быть представлена:

Заметим, что для упрощения решения можно заменить в оценке на . Сходимость оценки

обеспечит наличие мнимых полюсов объекта управления в числителе , т.е. в полиноме , что, учитывая (2.8), приведёт к отсутствию подобных полюсов в числителе регулятора.

Приведём общий вид используемого критерия:

Тогда уравнение Винера-Хопфа примет вид:

Выпишем в явном виде:

Для того, чтобы проанализировать влияние на получающийся регулятор и свойства системы, было реализовано автоматизированное решение уравнения Винера-Хопфа с , записанными выше (Приложение А, Листинг 2). Предполагается, что, при должной настройке, эти функции можно будет использовать для анализа подобных задач и их решений – основная проблема конструирования подобного комплекса инструментов заключается в распределении нулевых полюсов и нулей при факторизации и сепарации.

Приведём решение, полученное при :

Найдём корни :

Наличие полюсов объекта управлении в разности сигнализирует о правильности решения – при вычислении соответствующие корни следует сократить.

Найдём теперь корни характеристического полинома сконструированной системы:

Можно утверждать, что система устойчива – все корни характеристического полинома лежат в левой полуплоскости. Степень же устойчивости равна .

Схема модели аналогична схеме на рис. 5.

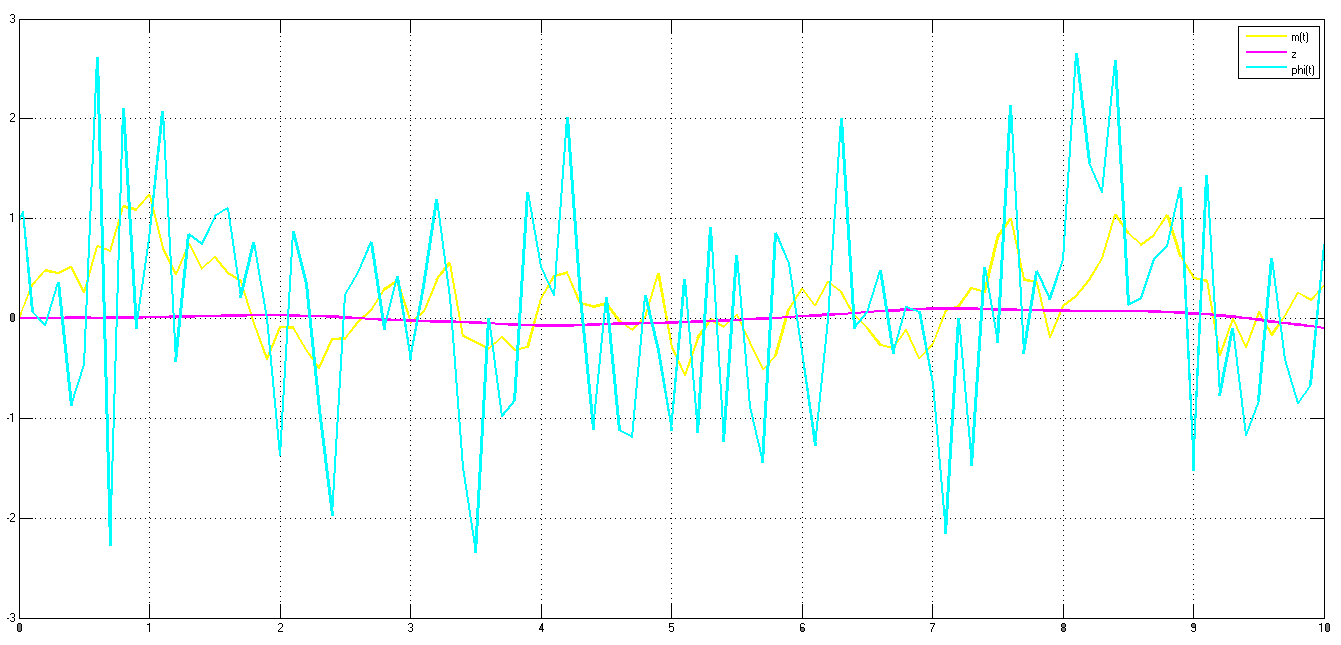


Рисунок 9: График слежения №3

Приведём теперь графики, с помощью которых удобно анализировать влияние значения веса на сконструированную систему.

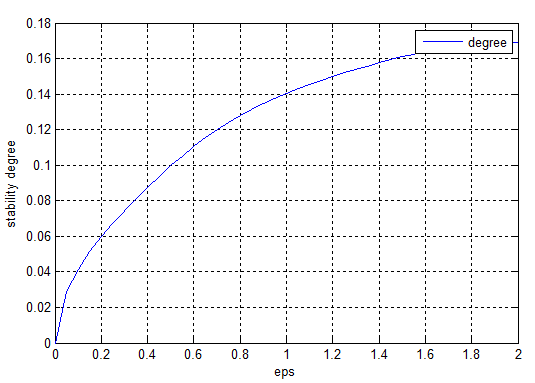


Рисунок 0: График зависимости степени устойчивости системы от выбора

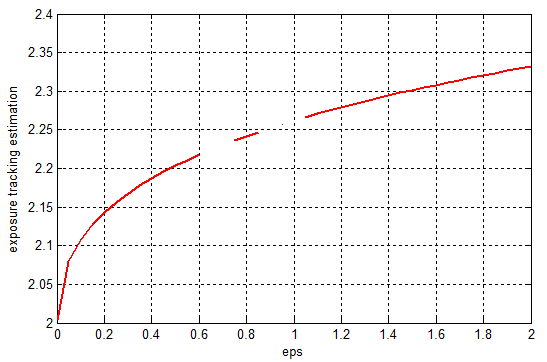


Рисунок 11: График зависимости значения оценки качества слежения от

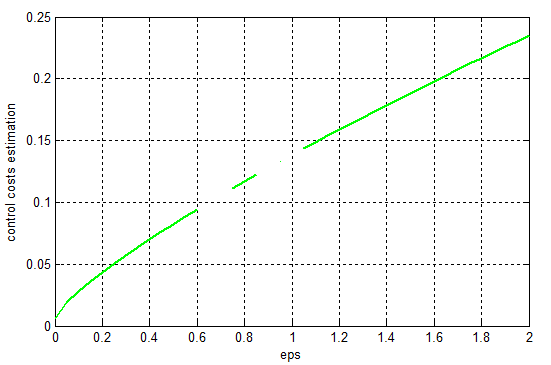


Рисунок 1: График зависимости значения оценки мощности управления от

Графики были построены поточечно с шагом в . Разрывы графика появлялись там, где значения соответствующих оценок, т.е.

для рис. 11 и

для рис. 12 принимали значения . Проблема появления таких значений заслуживает отдельного исследования и лежит вне рамок данной работы. В качестве вероятных причин можно назвать неспособность Matlab посчитать эти интегралы аналитически, либо же наличие полюсов в пределах интегрирования, приводящих к расхождению интеграла. Тогда, возможно, имеет смысл найти главные значения соответствующих интегралов. Также стоит в указанных случаях вычислить индикатор совместимости исходных данных Винера-Хопфа.

Продолжим анализ построенных графиков. Как видно из рис. 10 мы можем улучшить степень устойчивости с помощью увеличения веса . Но подобный выигрыш требует своей цены – из рис. 11-12 видно, что с ростом веса ухудшаются желаемые качества системы, сформулированные в постановке.

Таким образом было установлено, что функционал вида (2.13) позволяет решить поставленную задачу в пространстве операторов, а также даёт поле для маневрирования при нахождении регулятора – конструктор, используя графики на рис. 10-12, может, выбирая значение , получить систему с желаемыми оптимальными в силу каких-либо условий параметрами.

*Пространство состояний*

Для того, чтобы реализовать добавку (2.12) необходимо расширить фазовое пространство новыми переменными.

Для этого нам надо ввести такую переменную

Обратимся к рис. 4. Нам надо дополнить схему так, чтобы у нас появился такой выход , что передаточная функция для него относительно входа будет иметь вид:

|  |  |
| --- | --- |
|  | (3.2) |

Запишем выражение относительно сигнала и желаемого выхода :

Чтобы получить соотношение (3.2) следует пропустить сигнал через звено . Найдём его реализацию в пространстве состояний:

Из постановки . Тогда

Найдём теперь матрицу передаточных функций системы:

Отсюда следует, что в качестве надо выбрать .

Приведём теперь структурную схему расширенной системы:

Рисунок 13: Структурная схема расширенной системы

Новый критерий примет вид:

Запишем теперь все данные, необходимые для решения задачи с помощью функции build\_regulator7dim (Приложение А Листинг 1):

Проведём анализ свойств системы:

Условия, указанные в теореме, не выполняются.

Доведём решение до конца:

Выпишем передаточную функцию:

Построенный таким образом регулятор сам оказался неустойчив – его характеристический полином (т.е. ) содержит корни, лежащие на мнимой оси: . Модель для данного случая не конструировалась в силу своей громоздкости.

Таким образом было установлено, что использование функционала (2.13), т.е. реализация добавки (2.12) через расширение фазового пространства, не позволило решить задачу в пространстве оригиналов.

* + 1. **Третий способ: функционал с ограничением на компенсацию полюсов, реализация с помощью воздействия вида белого шума.**

Воспользуемся теперь критерием (2.14).

*Пространство состояний*

Структурная схема системы с учётом введённого воздействия примет вид:

Рисунок 14: Структурная схема системы с введённым фиктивным воздействием

Запишем соответствующее ей описание:

Положим и проведём анализ её свойств:

Условия теоремы выполняются.

С помощью функции build\_regulator были найдены:

Передаточная функция регулятора приняла вид:

Корни характеристического уравнения:

Т.о. задача была решена в пространстве состояний только с помощью критерия вида (2.14).

*Пространство операторов*

Проанализируем особенности второго способа в пространстве изображений.

Критерий в данной случае запишется следующим образом:

Заметим, что последнюю добавку можно откинуть – при решении в пространстве операторов она не несёт никакой нагрузки. При решении же в пространстве состояний она добавляется автоматически – избавиться от неё невозможно – и способствует искажению постановки задачи.

в данной постановке совпадёт с из предыдущего примера. А можно выразить через :

Построенный регулятор принял вид:

Т.о. критерий (2.14) тоже позволяет разрешить задачу в пространстве операторов.

Построим теперь модель, сравнивающую оба полученных регулятора:

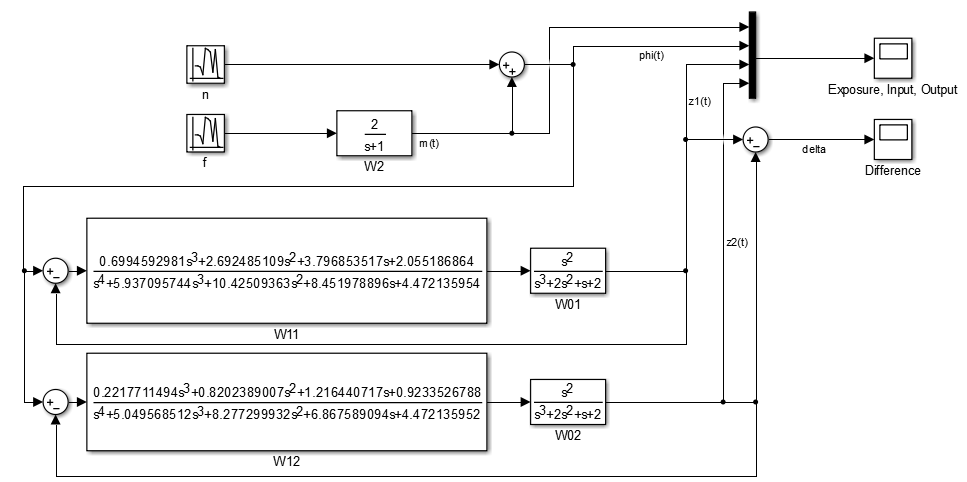


Рисунок 15: Построенная модель

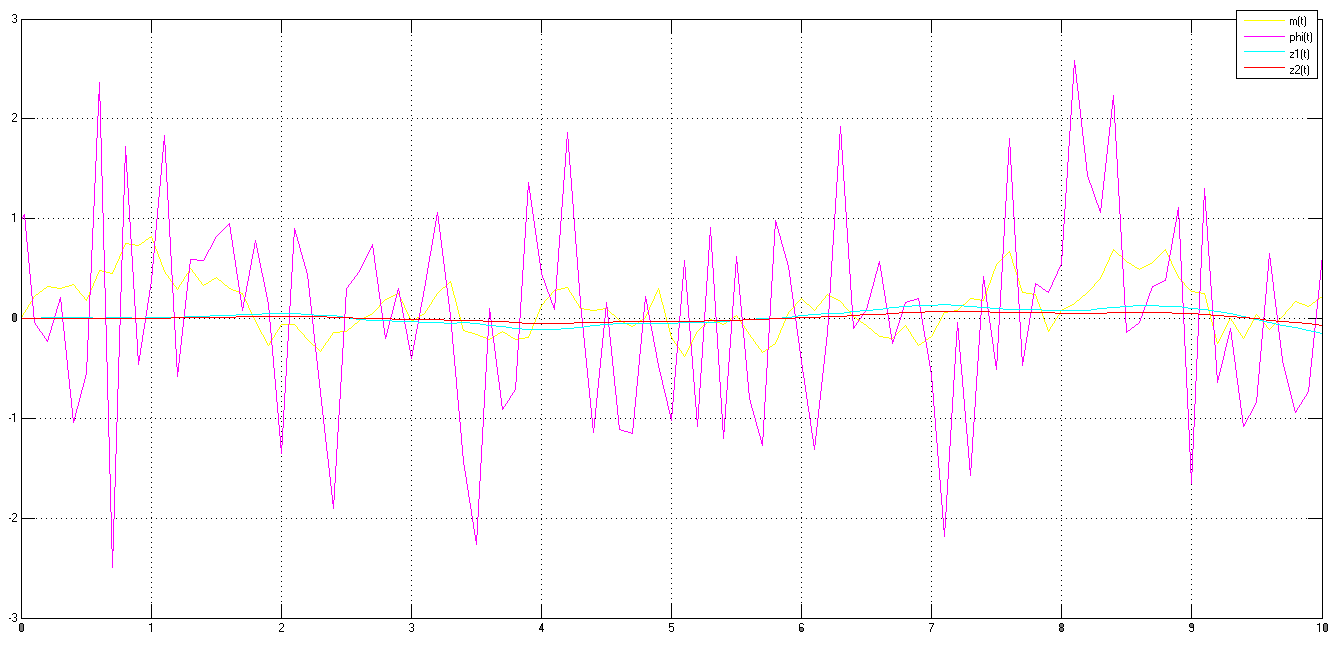


Рисунок 16: вывод элемента Exposure, Input, Output

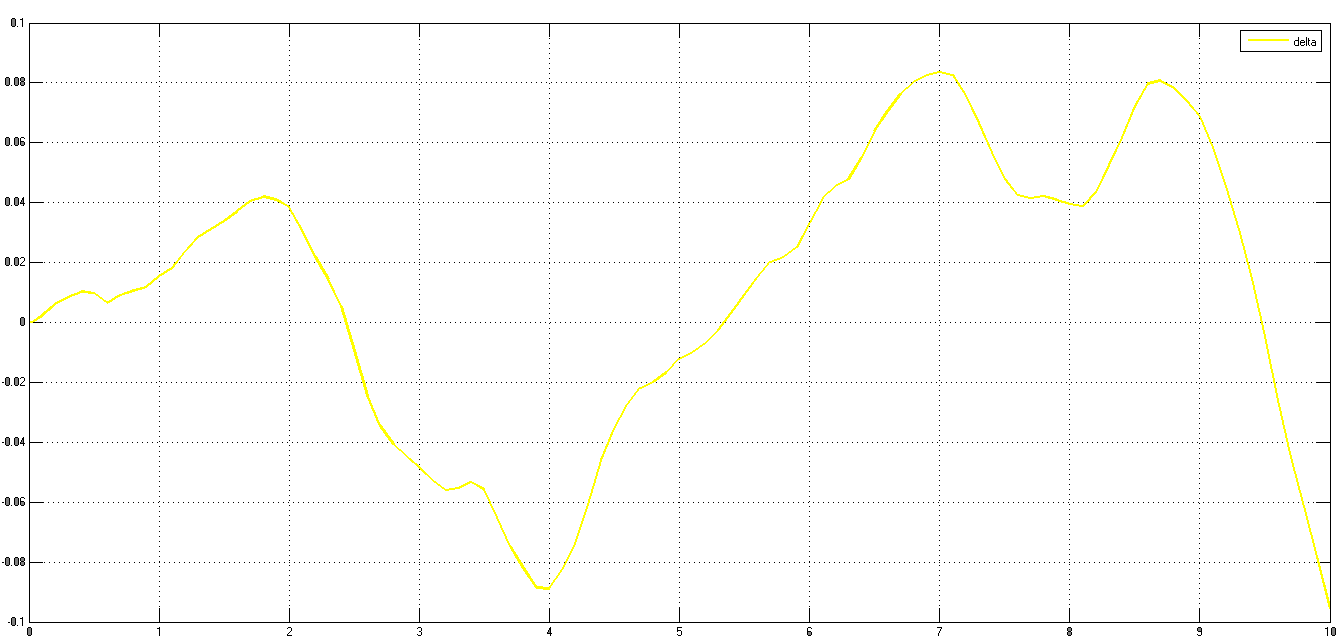


Рисунок 17: вывод элемента Difference

**Заключение**

В настоящей работе была проанализирована проблема оптимальной стабилизирующей фильтрации для класса объектов управления с полюсами на мнимой оси, был построен алгоритм конструирования функционала, который бы не только решал задачу воспроизведения задающего воздействия, но и придавал системе свойство устойчивости. Приведены критерии, позволяющие решить задачу как в пространстве состояний, так и в пространстве операторов. Обозначен механизм увеличения степени устойчивости системы с помощью выбора в критерии веса при оценке, ограничивающей компенсацию регулятором мнимых полюсов объекта управления. В среде Matlab разработаны два набора инструментов, позволяющих строить фильтр Калмана и находить решение поставленной задачи в множестве управляющий устройств с недостаточным числом звеньев коррекции. Также было проведено моделирование сконструированных систем с помощью пакета Simulink, представлены графики, характеризующие качество слежение системы.

Адекватность приведённого алгоритма подкреплена практическим решением задачи, обосновывающим теоретические выводы. Сравнительный анализ хода решения и его результатов в пространстве состояний и операторов позволяет утверждать, что задачу в пространстве оригиналов можно решить только с помощью введения фиктивного воздействия, что может ухудшить качества получаемого регулятора, так как в функционале появляется добавка, не имеющая практической ценности.

**Список литературы**

1. Дорф. Р, Бишоп Р. Пер. с англ. Копылова Б.И. Современные системы управления. – М.: Лаборатория Базовых Знаний, 2002. – 832 с.:ил.
2. Зотов М.Г. Многокритериальное конструирование систем автоматического управления. – М.: БИНОМ. Лаборатория знаний, 2004. – 375 с., ил.
3. Зотов М.Г. О негрубости систем управления для некоторого класса объектов и способах её устранения. // Автоматика и телемеханика, 2009, №3, С.34-47.
4. Первозванский А.А. Курс теории автоматического управления: Учеб. пособие – М.: Наука. Гл. ред. физ.-мат. лит., 1986. – 616 с.
5. Поляк Б.Т., Щербаков П.С. Робастная устойчивость и управление. – М.: Наука, 202. – 303 с.

**Приложение A**

Листинг 1

*build\_regulator.m*

function [K, L, W] = build\_regulator(A,B,G,C,Q,R,F,V,PHI,mode)

%by setting mode you can select way of L finding

function F = rik\_eval(z)

F = zeros(1,16);

count = 1;

d11=sym('d11');d12=sym('d12');d13=sym('d13');d14=sym('d14');d21=sym('d21');d22=sym('d22');d23=sym('d23');d24=sym('d24');

d31=sym('d31');d32=sym('d32');d33=sym('d33');d34=sym('d34');d41=sym('d41');d42=sym('d42');d43=sym('d43');d44=sym('d44');

for l=1:1:4

for j=1:1:4

F(count) = subs(RES(l,j),[d11,d12,d13,d14,d21,d22,d23,d24,d31,d32,d33,d34,d41,d42,d43,d44],z);

count = count + 1;

end

end

end

dim = size(A, 1) - 1;

H = C\*B;

for i=1:1:dim

H = H \* C\*A^(dim)\*B;

end

fprintf('(A,C) = %d, (A,B) = %d, (A,G) = %d, (C,A,B) = %d\n', rank(obsv(A,C)), rank(ctrb(A,B)), rank(ctrb(A,G)), rank(H));

fprintf('K coeff:\n');

K = lqr(A,B,Q,R)

if mode == 0

fprintf('Finding L with lqe:\n');

if PHI == 0

L = lqe(A,G,C,F,V);

else

L = lqe(A,G,C,F,V,PHI);

end

else

if mode == 1

fprintf('D system for L finding:\n');

syms d11;syms d12;syms d13;syms d14;syms d21;syms d22;syms d23;syms d24;syms d31;syms d32;syms d33;syms d34;syms d41;syms d42;syms d43;syms d44;

D\_SYM=[d11 d12 d13 d14;d21 d22 d23 d24;d31 d32 d33 d34;d41 d42 d43 d44];

RES = A\*D\_SYM+D\_SYM\*A'+G\*F\*G'-(D\_SYM\*C'+G\*PHI)\*V^(-1)\*(D\_SYM\*C'+G\*PHI)'

fprintf('Numerically solving system of equations now:\n');

%options = optimoptions('fsolve','TolFun', 1e-18,'TolX', 1e-18);

x = fsolve(@rik\_eval, zeros(1, 16));

D = [x(1:4);x(5:8);x(9:12);x(13:16)]

end

if mode == 2

fprintf('Analitacly solving system of equations now:\n');

x = solve(RES(1,1),RES(1,2),RES(1,3),RES(1,4),RES(2,1),RES(2,2),RES(2,3),RES(2,4),RES(3,1),RES(3,2),RES(3,3),RES(3,4),RES(4,1),RES(4,2),RES(4,3),RES(4,4),d11,d12,d13,d14,d21,d22,d23,d24,d31,d32,d33,d34,d41,d42,d43,d44);

D = [x.d11,x.d12,x.d13,x.d14;x.d21,x.d22,x.d23,x.d24;x.d31,x.d32,x.d33,x.d34;x.d41,x.d42,x.d43,x.d44]

end

L = (D\*C'+G\*PHI)\*V^(-1);

end

fprintf('Found L:'); L

fprintf('Finding W(s):\n');

X1=sym('X1'); X2=sym('X2'); X3=sym('X3'); X4=sym('X4'); U=sym('U'); Y=sym('Y');

X\_IZ = [X1;X2;X3;X4];

s=sym('s');

EQ1 = s\*X\_IZ - A\*X\_IZ - B\*U - L\*(Y - C\*X\_IZ);

EQ2 = U + K\*X\_IZ;

RESULT = solve(EQ1(1,1),EQ1(2,1),EQ1(3,1),EQ1(4,1),EQ2,U,X1,X2,X3,X4);

W = RESULT.U

End

*build\_regulator7dim.m*

function [K, L, W] = build\_regulator7dim(A,B,G,C,Q,R,F,V,PHI,mode)

function F = rik\_eval(z)

F = zeros(1,16);

count = 1;

d11=sym('d11');d12=sym('d12');d13=sym('d13');d14=sym('d14');d15=sym('d15');d16=sym('d16');d17=sym('d17');

d21=sym('d21');d22=sym('d22');d23=sym('d23');d24=sym('d24');d25=sym('d25');d26=sym('d26');d27=sym('d27');

d31=sym('d31');d32=sym('d32');d33=sym('d33');d34=sym('d34');d35=sym('d35');d36=sym('d36');d37=sym('d37');

d41=sym('d41');d42=sym('d42');d43=sym('d43');d44=sym('d44');d45=sym('d45');d46=sym('d46');d47=sym('d47');

d51=sym('d51');d52=sym('d52');d53=sym('d53');d54=sym('d54');d55=sym('d55');d56=sym('d56');d57=sym('d57');

d61=sym('d61');d62=sym('d62');d63=sym('d63');d64=sym('d64');d65=sym('d65');d66=sym('d66');d67=sym('d67');

d71=sym('d71');d72=sym('d72');d73=sym('d73');d74=sym('d74');d75=sym('d75');d76=sym('d76');d77=sym('d77');

sub\_array = [d11,d12,d13,d14,d15,d16,d17,d21,d22,d23,d24,d25,d26,d27,d31,d32,d33,d34,d35,d36,d37,d41,d42,d43,d44,d45,d46,d47,d51,d52,d53,d54,d55,d56,d57,d61,d62,d63,d64,d65,d66,d67,d71,d72,d73,d74,d75,d76,d77];

for l=1:1:4

for j=1:1:4

F(count) = subs(RES(l,j),sub\_array,z);

count = count + 1;

end

end

end

dim = size(A, 1) - 1;

H = C\*B;

for i=1:1:dim

H = H \* C\*A^(dim)\*B;

end

fprintf('(A,C) = %d, (A,B) = %d, (A,G) = %d, (C,A,B) = %d\n', rank(obsv(A,C)), rank(ctrb(A,B)), rank(ctrb(A,G)), rank(H));

fprintf('K coeff:\n');

K = lqr(A,B,Q,R)

if mode == 0

fprintf('Finding L with lqe:\n');

if PHI == 0

L = lqe(A,G,C,F,V);

else

L = lqe(A,G,C,F,V,PHI);

end

else

if mode == 1

fprintf('D system for L finding:\n');

d11=sym('d11');d12=sym('d12');d13=sym('d13');d14=sym('d14');d15=sym('d15');d16=sym('d16');d17=sym('d17');

d21=sym('d21');d22=sym('d22');d23=sym('d23');d24=sym('d24');d25=sym('d25');d26=sym('d26');d27=sym('d27');

d31=sym('d31');d32=sym('d32');d33=sym('d33');d34=sym('d34');d35=sym('d35');d36=sym('d36');d37=sym('d37');

d41=sym('d41');d42=sym('d42');d43=sym('d43');d44=sym('d44');d45=sym('d45');d46=sym('d46');d47=sym('d47');

d51=sym('d51');d52=sym('d52');d53=sym('d53');d54=sym('d54');d55=sym('d55');d56=sym('d56');d57=sym('d57');

d61=sym('d61');d62=sym('d62');d63=sym('d63');d64=sym('d64');d65=sym('d65');d66=sym('d66');d67=sym('d67');

d71=sym('d71');d72=sym('d72');d73=sym('d73');d74=sym('d74');d75=sym('d75');d76=sym('d76');d77=sym('d77');

D\_SYM=[d11,d12,d13,d14,d15,d16,d17;d21,d22,d23,d24,d25,d26,d27;d31,d32,d33,d34,d35,d36,d37;d41,d42,d43,d44,d45,d46,d47;d51,d52,d53,d54,d55,d56,d57;d61,d62,d63,d64,d65,d66,d67;d71,d72,d73,d74,d75,d76,d77];

RES = A\*D\_SYM+D\_SYM\*A'+G\*F\*G'-(D\_SYM\*C'+G\*PHI)\*V^(-1)\*(D\_SYM\*C'+G\*PHI)'

fprintf('Numerically solving system of equations now:\n');

%options = optimoptions('fsolve','TolFun', 1e-18,'TolX', 1e-18);

x = fsolve(@rik\_eval, zeros(1, 49));

D = [x(1:7);x(8:14);x(15:21);x(22:28);x(29:35);x(36:42);x(43:49)]

end

L = (D\*C'+G\*PHI)\*V^(-1);

end

fprintf('Found L:'); L

fprintf('Finding W(s):\n');

X1=sym('X1'); X2=sym('X2'); X3=sym('X3'); X4=sym('X4'); X5=sym('X5'); X6=sym('X6'); X7=sym('X7'); U=sym('U'); Y=sym('Y');

X\_IZ = [X1;X2;X3;X4;X5;X6;X7];

s=sym('s');

EQ1 = s\*X\_IZ - A\*X\_IZ - B\*U - L\*(Y - C\*X\_IZ);

EQ2 = U + K\*X\_IZ;

RESULT = solve(EQ1(1,1),EQ1(2,1),EQ1(3,1),EQ1(4,1),EQ1(5,1),EQ1(6,1),EQ1(7,1),EQ2,U,X1,X2,X3,X4,X5,X6,X7);

W = RESULT.U

end

Листинг 2

*stablility\_degree.m*

function [ degree, est1, est2 ] = stability\_degree( eps )

if size(eps, 1) > 1

length = size(eps, 1);

degree = zeros(length, 1);

est1 = zeros(length, 1);

est2 = zeros(length, 1);

for i=1:1:size(eps)

fprintf('eps = %f\n', eps(i));

[degree(i, 1), est1(i, 1), est2(i,1)] = stability\_degree( eps(i) );

fprintf('degree = %f, est1 = %f, est2 = %f\n', degree(i, 1), est1(i, 1), est2(i,1));

end

return

end

% S = Q/P

s = sym('s');

W0Q = s^2;

W0P = (s^3+2\*s^2+s+2);

S1Q = s^14 - 10\*s^12 + 17\*s^10 + 67\*s^8 + s^6\*(eps - 66) - s^4\*(eps + 289) - 264\*s^2 - 80;

S1P = s^4 \* (1 - s^2)\*(s^3 + 2\*s^2 + s + 2)\*(s^3 - 2\*s^2 + s - 2);

S2Q = 4\*s^6 - 8\*s^4 + s^2\*(eps - 28) - eps - 16;

S2P = (1 - s^2)\*(s^3 + 2\*s^2 + s + 2)\*(s^3 - 2\*s^2 + s - 2);

%distributing roots

S1Q\_roots = solve(S1Q)

S1P\_roots = solve(S1P)

[S1Q\_left\_roots, S1Q\_right\_roots] = distribute\_roots(S1Q\_roots)

[S1P\_left\_roots, S1P\_right\_roots] = distribute\_roots(S1P\_roots)

S1Q\_mult = get\_multiplier(S1Q)

S1P\_mult = get\_multiplier(S1P)

S1Q\_right = sym\_from\_roots(S1Q\_right\_roots, S1Q\_mult)

S1P\_right = sym\_from\_roots(S1P\_right\_roots, S1P\_mult)

fprintf('###S3 = S2 / S1-###\n');

S3Q = S2Q \* S1P\_right

S3P = S2P \* S1Q\_right

fprintf('simplified fraction:\n');

[S3Q, S3P] = simplify\_sym\_fraction(S3Q, S3P)

numerator = sym2poly(S3Q)

num\_mult = numerator(1)

denominator = sym2poly(S3P)

den\_mult = denominator(1)

numerator = num\_mult^(-1) \* numerator;

denominator = den\_mult^(-1) \* denominator;

[residues, poles, direct] = residue(numerator, denominator)

count = 1;

for i=1:1:size(poles)

real\_part = real(poles(i));

if abs(real\_part) < 10e-12

real\_part = 0;

end

if real\_part <= 0

new\_poles(count) = poles(i);

new\_residues(count) = residues(i);

count = count + 1;

end

end

new\_poles

new\_residues

[new\_numerator, new\_denominator] = residue(new\_residues, new\_poles, 0);

new\_numerator = num\_mult \* new\_numerator

new\_denominator = den\_mult \* new\_denominator

fprintf('###S4 = [S2 / S1-]+###\n');

S4Q = sym\_from\_poly(new\_numerator)

S4P = sym\_from\_poly(new\_denominator)

fprintf('###HV = S4/S1+###\n');

S1Q\_left = sym\_from\_roots(S1Q\_left\_roots, 1)

S1P\_left = sym\_from\_roots(S1P\_left\_roots, 1)

HV\_Q = S4Q \* S1P\_left

HV\_P = S4P \* S1Q\_left

fprintf('simplified fraction:\n');

[HV\_Q, HV\_P] = simplify\_sym\_fraction(HV\_Q, HV\_P)

fprintf('###HV\_Q roots###\n');

solve(HV\_Q)

fprintf('###HV\_P - HV\_Q roots###\n');

solve(HV\_P - HV\_Q)

W1Q = (W0P \* HV\_Q);

W1P = (W0Q \* (HV\_P - HV\_Q));

fprintf('simplified fraction:\n');

[W1Q, W1P] = simplify\_sym\_fraction(W1Q, W1P)

W1Q\_roots = solve(W1Q);

W1P\_roots = solve(W1P);

CP\_roots = solve(W1Q\*W0Q + W1P\*W0P)

d = 100;

for i=1:1:size(CP\_roots)

if real(CP\_roots(i)) >= 0

fprintf('UNSTABLE ROOT:');

disp(CP\_roots(i));

fprintf('\n');

degree = 0;

return

end

if abs(real(d)) > abs(real(CP\_roots(i)))

d = CP\_roots(i);

end

end

root = d;

degree = abs(real(d));

HV = HV\_Q / HV\_P;

HV\_MIN = subs(HV, s, -s);

W0 = W0Q / W0P;

W0\_MIN = subs(W0, s, -s);

S\_mm = 4/(1-s^2);

S\_nn = 1;

EST1 = 1/(2\*1i\*pi) \* int(HV\*HV\_MIN\*(S\_mm + S\_nn) - (HV-HV\_MIN)\*S\_mm + S\_mm, s, -1i\*inf, 1i\*inf);

EST2 = 1/(2\*1i\*pi) \* int(W0^(-1)\*W0\_MIN^(-1)\*HV\*HV\_MIN\*S\_mm, s, -1i\*inf, 1i\*inf);

est1 = double(EST1);

est2 = double(EST2);

*simplify\_sym\_fraction.m*

function [ new\_num, new\_den ] = simplify\_sym\_fraction( num, den )

s = sym('s');

num\_mult = get\_multiplier(num)

num\_roots = solve(num);

den\_mult = get\_multiplier(den)

den\_roots = solve(den);

[new\_num\_roots, new\_den\_roots] = delete\_intersected\_roots(num\_roots, den\_roots);

new\_num = sym\_from\_roots(new\_num\_roots, num\_mult/den\_mult);

new\_den = sym\_from\_roots(new\_den\_roots, 1);

end

*delete\_intersected\_roots.m*

%not optimized, but pretty recoursed!

function [ roots1, roots2 ] = delete\_intersected\_roots( roots1, roots2 )

i = 0; j = 0;

for i=1:1:size(roots1)

for j=1:1:size(roots2)

% this is dangerous place. normally, we can't delete near roots

% if they are from right plane.

if (roots1(i) == roots2(j)) || (abs(roots1(i)-roots2(j)) < 10e-12)

roots1(i) = [];

roots2(j) = [];

[ roots1, roots2 ] = delete\_intersected\_roots(roots1, roots2);

return

end

end

end

end

*distribute\_roots.m*

function [left\_roots, right\_roots] = distribute\_roots(roots\_arr)

l\_count = 0;

r\_count = 0;

poles\_count = 0;

for j=1:1:size(roots\_arr)

if real(roots\_arr(j)) < 0

l\_count = l\_count + 1;

left\_roots(l\_count) = roots\_arr(j);

elseif real(roots\_arr(j)) > 0

r\_count = r\_count + 1;

right\_roots(r\_count) = roots\_arr(j);

else

poles\_count = poles\_count + 1;

poles\_arr(poles\_count) = roots\_arr(j);

end

end

if l\_count == 0

left\_roots = [];

end

if r\_count == 0

right\_roots = [];

end

if poles\_count > 0

[left\_poles, right\_poles] = distribute\_poles(poles\_arr);

left\_roots = (cat(2, left\_poles, left\_roots));

right\_roots = (cat(2, right\_poles, right\_roots));

end

left\_roots = left\_roots';

right\_roots = right\_roots';

end

function [counted\_poles] = add\_pole(counted\_poles, pole)

found = 0;

for i=1:1:size(counted\_poles)

if counted\_poles(i,1) == pole

counted\_poles(i,2) = counted\_poles(i,2) + 1;

found = 1;

break

end

end

if found ~= 1

poles\_count = size(counted\_poles, 1) + 1;

counted\_poles(poles\_count,1) = pole;

counted\_poles(poles\_count,2) = 1;

end

end

%we distribute poles equally between both planes

function [left\_poles, right\_poles] = distribute\_poles(poles)

counted\_poles = [];

left\_poles = [];

right\_poles = [];

for j=1:1:size(poles, 2)

[counted\_poles] = add\_pole(counted\_poles, poles(1,j));

end

left\_poles\_count = 1;

right\_poles\_count = 1;

for j=1:1:size(counted\_poles)

for k=1:1:(counted\_poles(j,2)/2)

left\_poles(left\_poles\_count) = counted\_poles(j,1);

left\_poles\_count = left\_poles\_count + 1;

right\_poles(right\_poles\_count) = counted\_poles(j,1);

right\_poles\_count = right\_poles\_count + 1;

end

end

end

*get\_multiplier.m*

function [multiplier] = get\_multiplier(sym\_poly)

poly = sym2poly(sym\_poly);

multiplier = poly(1);

end

*sym\_from\_poly.m*

function [sym\_poly] = sym\_from\_poly(poly)

s = sym('s');

sym\_poly = 0;

degree = size(poly, 2);

for k=1:1:degree

real\_part = real(poly(k));

imag\_part = imag(poly(k));

if abs(imag\_part) < 10e-12

imag\_part = 0;

end

if abs(real\_part) < 10e-12

real\_part = 0;

end

coeff = real\_part + imag\_part \* 1i;

sym\_poly = sym\_poly + coeff\*s^(degree-k);

end

end

*sym\_from\_roots.m*

function [ sym\_poly ] = sym\_from\_roots( roots, multiplier)

s = sym('s');

sym\_poly = 1;

for k=1:1:size(roots)

sym\_poly = sym\_poly \* (s - roots(k));

end

sym\_poly = multiplier \* sym\_poly;

end