# 3. Проектирование САУ

## 3.1. Функциональная схема устройства

В данной работе рассмотрим только режим ручного управления с возможностью подъема в транспорт, так как это основной режим работы данного устройства. За данный режим работы отвечает многопозиционный джойстик. Функциональная схема разрабатываемого устройства показана на Рис.3.1.

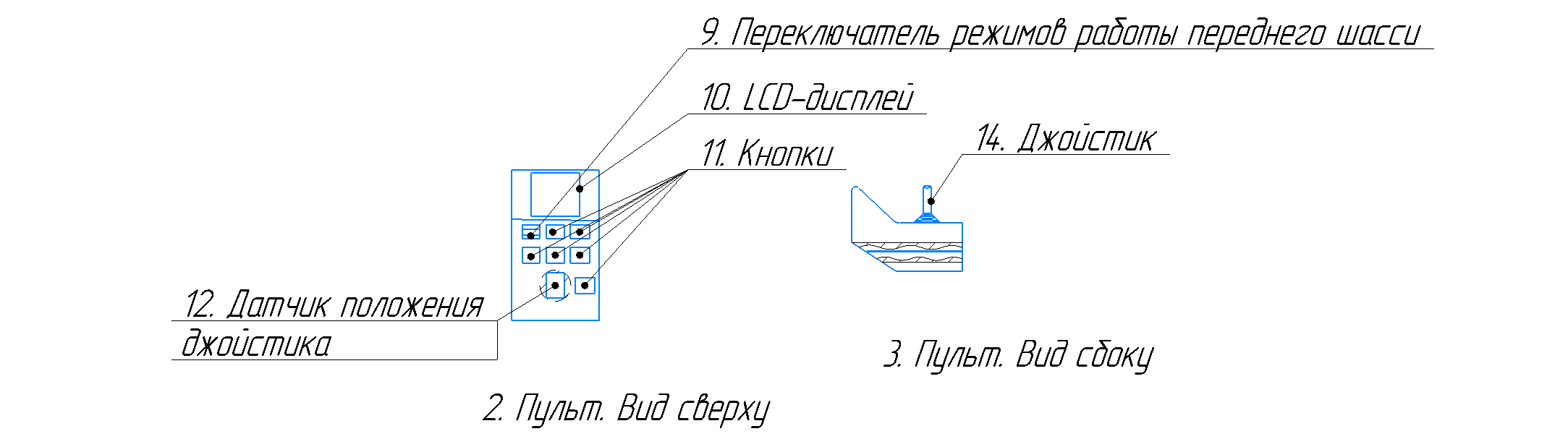
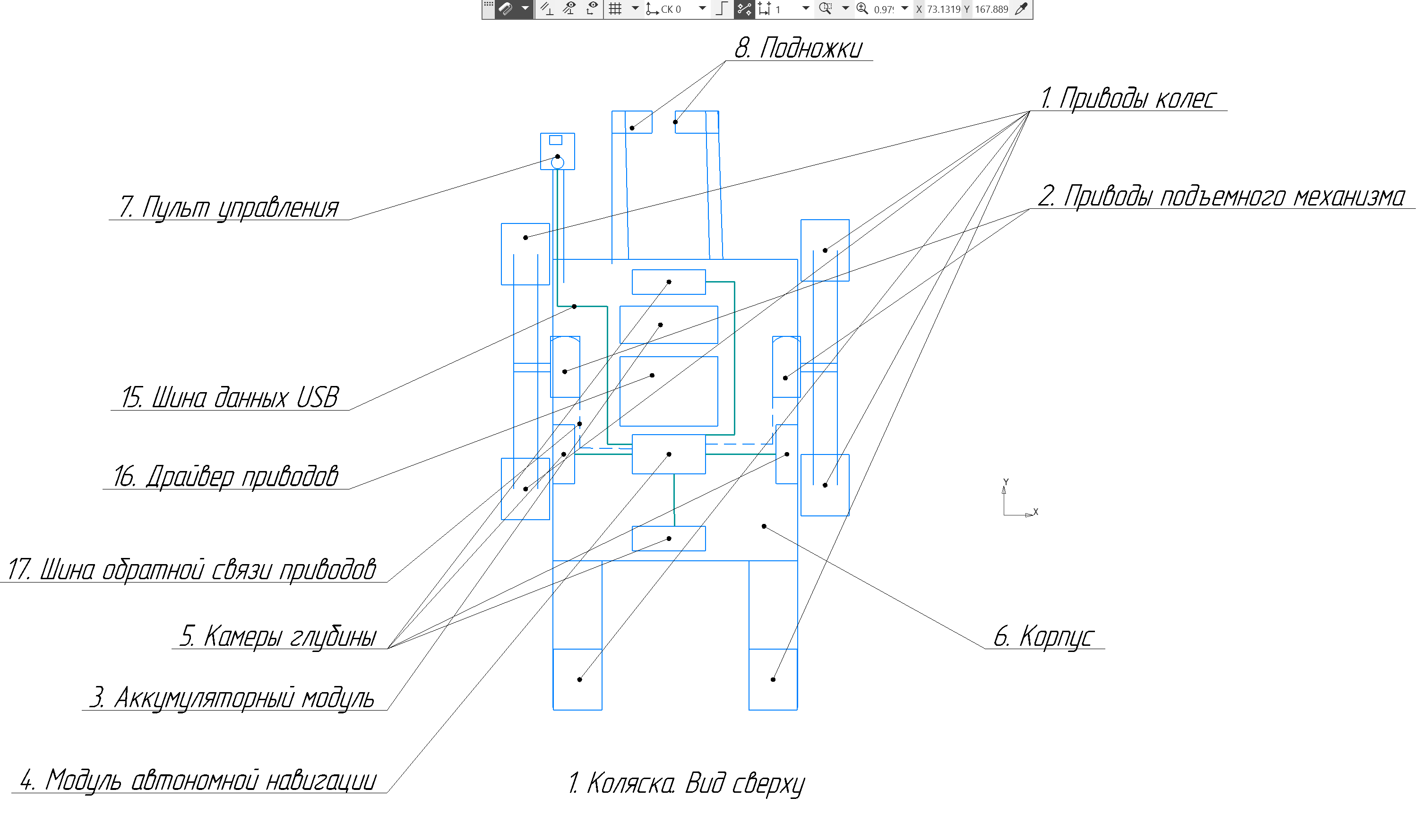


Рис. 3.1. Функциональная схема коляски

В рассматриваемом режиме работы управление движением осуществляется с помощью пульта ручного управления 7. Пульт считывает магнитным датчиком 12 положение джойстика 14 преобразует полученные данные в значения ШИМ-сигналов, которые необходимо подать на приводы колес. Затем пульт отправляет вычисленные значения в драйвер приводов 16 через модуль автономной навигации 4, который при наличии препятствия спереди на расстоянии, меньшем чем 20 см, останавливает движение. Расстояние до препятствий определяется по показаниям камер глубины 5. Пульт ручного управления, камеры глубины и модуль автономной навигации соединены USB-шиной 15. На пульте кроме джойстика размещены кнопки управления устройством 11 и LCD-дисплей 10, а также трехпозиционный переключатель 9 управления режимом работы приводов подъемного механизма 2. На LCD-дисплее отображаются данные о текущем режиме работы устройства, его скорости и уровне заряда аккумуляторов. Скорость коляски вычисляется по данным с приводов колес 1, а уровень заряда считывается с датчика напряжения в аккумуляторном модуле 3. Данные о текущей скорости приводов колес передаются в модуль автономной навигации по шине информации обратной связи 17, также как и угол отклонения двигателей подъема шасси.

С помощью переключателя 9 можно включить один из трех режимов работы приводов подъема шасси:

1. режим свободного вращения, когда приводы отключены от питания, поэтому шасси свободно вращаются;
2. режим подъема шасси, при котором приводы поднимают шасси на заданный угол над землей;
3. режим подъема центральных колес, при котором приводы поднимают центральные колеса на заданный угол, вращаясь в противоположном режиму подъема шасси направлении.

Алгоритм функционирования рассматриваемого режима такой:

1. пока переключатель 9 находится в положении 1, коляска двигается как обычное транспортное средство (развороты осуществляются по танковой схеме);
2. как только пассажир доезжает до входа в транспорт, то переводит переключатель 9 в режим 2, и, когда передние колеса поднимутся на заданный угол, с помощью джойстика наезжает ими на въезд;
3. после того, как передние колеса достаточно заедут на ступеньку входа, пассажир переводит переключатель 9 в режим 3, центральные колеса поднимаются, становясь на один уровень с передними, и пассажир с помощью джойстика заезжает в транспорт;
4. после заезда в транспорт пассажир переводит переключатель 9 в положение 1, снимая тем самым с переднего шасси ограничения движения.

## 3.2 Структурная схема САУ

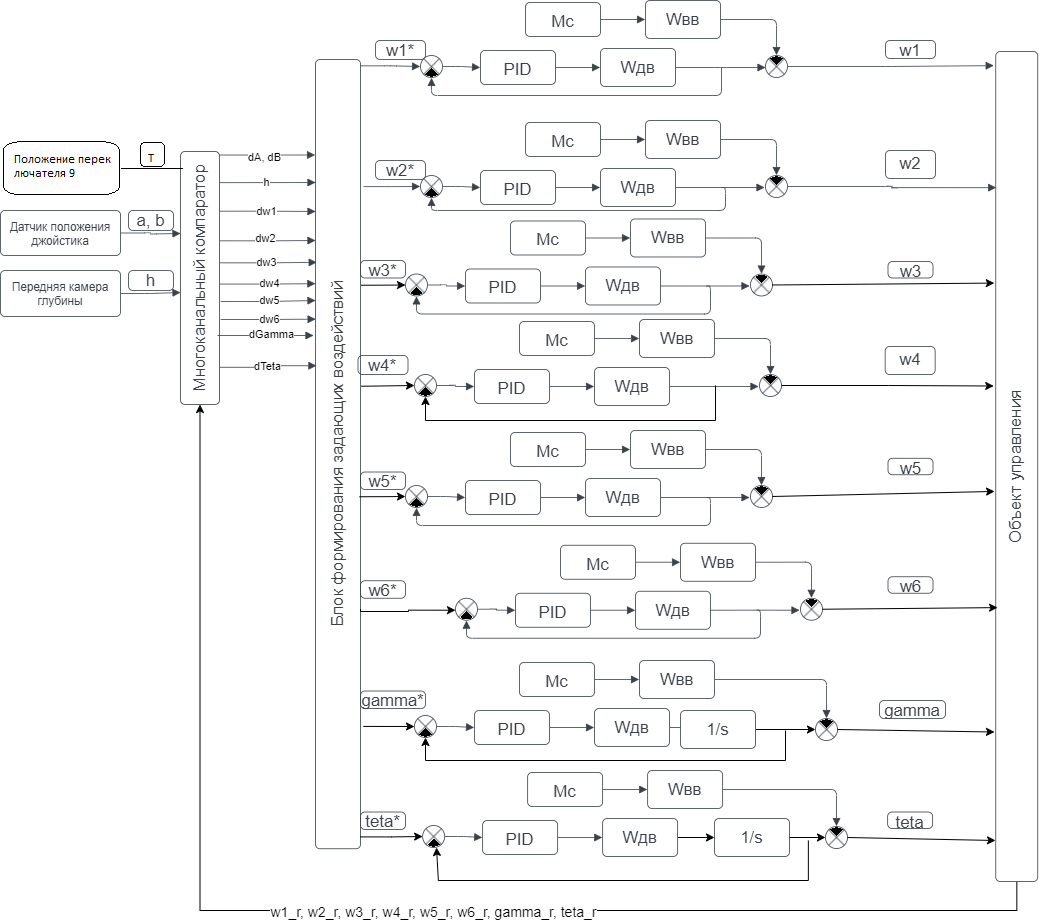
Рассмотрим структурную схему САУ в рассматриваемом режиме. Данная схема показана на рис.3.2. 

Рис.3.2. Структурная схема САУ в рассматриваемом режиме.

Первым блоком идет многоканальный компаратор, который, получая на вход положение переключателя 9 (на схеме обозначено как T), углы отклонения джойстика от вертикального положения (на схеме обозначены как a, b), расстояние до ближайшего препятствия h, а также данные о текущих скоростях колес () и положениях передних шасси () с энкодеров соответствующих приводов, вычисляет ошибки для скоростей колес и положений шасси (, ), и передает все эти данные в блок задающих воздействий. Блок задающих воздействий формирует желаемые значения скоростей колес и положений шасси (, ). Желаемые значения попадают в ветки управления отдельными приводами, на выходе которых мы получим скорости колес и положения шасси в следующий момент времени.

# 3.3 Настройка ПИД-регулятора привода колес

## 3.3.1 Настройка ПИД-регулятора САУ привода

Проектирование САУ устройства начнем с создания ПИД-регулятора для приводов колес, пока без учета нелинейности и робастности данных систем.

Для начала составим математическую модель электропривода. Математическая модель электромеханической системы (электропривода) описывается системой уравнений Максвелла-Лагранжа:

(3.1)

Заменим оператором Лапласа **р**:

(3.2)

Проведем следующие преобразования:

(3.3)

Подставим второе уравнение системы в первое:

(3.4)

Раскроем скобки:

(3.4)

Преобразуем выражение к виду:

(3.5)

Отсюда находим передаточные функции.

Передаточная функция двигателя:

(3.6)

Передаточная функция возмущающего воздействия:

(3.7)

Получим также из (3.5) дифференциальное уравнение для моделирования двигателя в Simulink:

(3.8)

Решим (3.8) относительно :

(3.9)

Блок-схема данного уравнения в Simulink показана на рис. 3.3

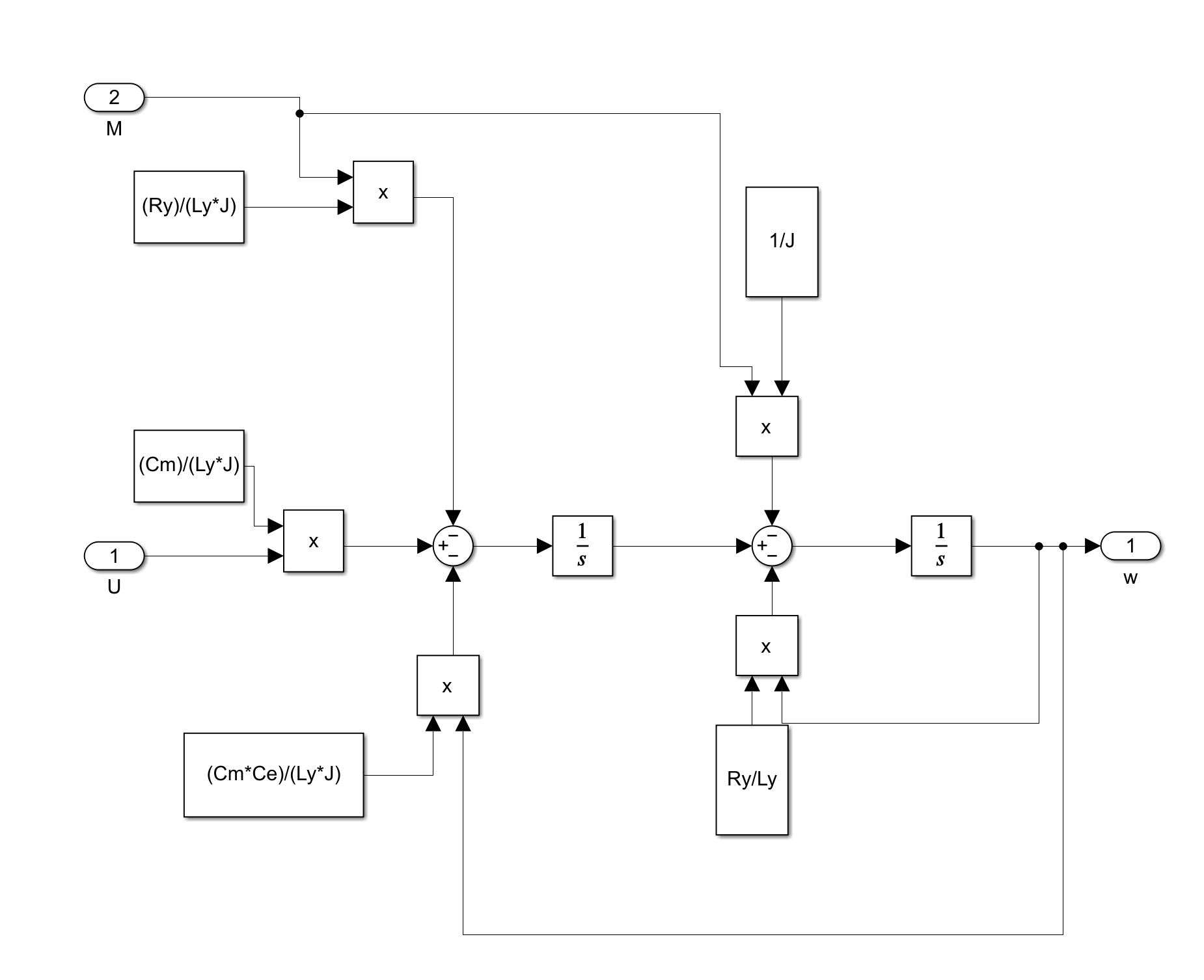


Рис.3.3. Блок-схема уравнения привода в Simulink

Данная блок-схема составлена в виде подсистемы мотора, на вход которой подается момент сопротивления двигателю M и напряжение питания U, выходом служит угловая скорость привода . Значения параметров двигателя зададим через командное окно MATLAB. С использованием данной подсистемы составим теперь блок схему оценки качества переходного процесса без ПИД-регулятора (рис.3.4) (момент сопротивления пока выставим не большим – 0.1 Н\*м).

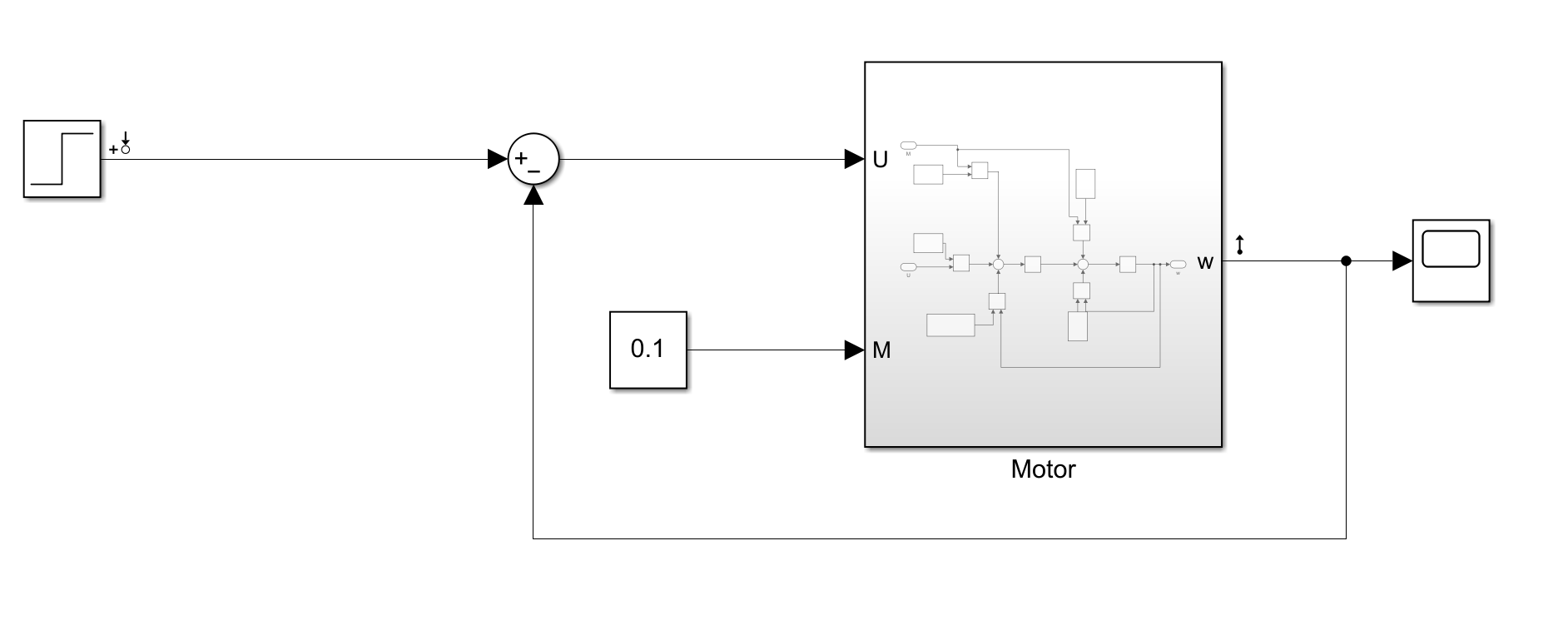


Рис.3.4. Блок-схема оценки качества переходного процесса

Выведем с помощью Model Linearizer текущие параметры переходного процесса рис.3.5.

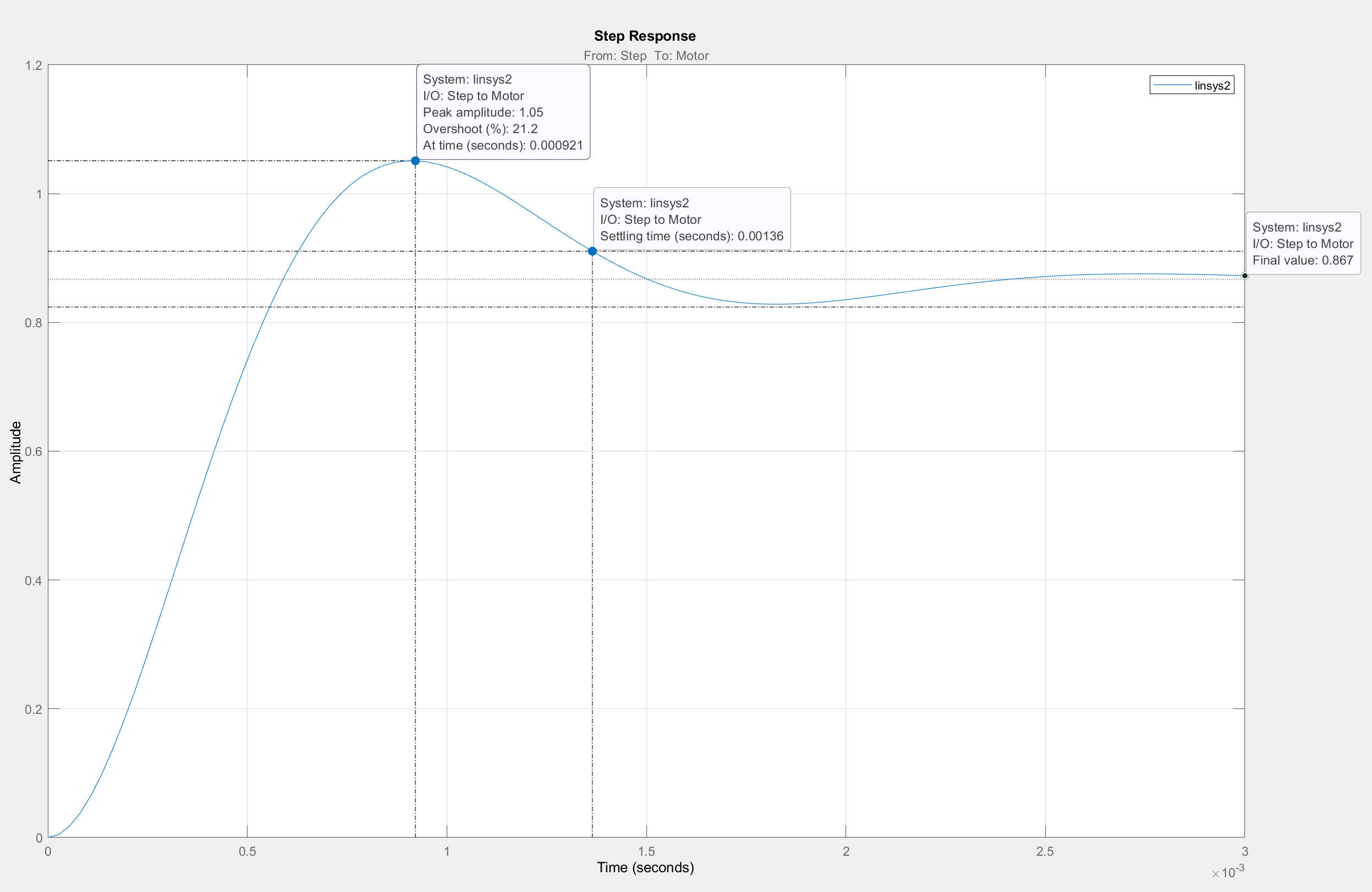


Рис.3.5. Характеристика переходного процесса

Перерегулирование в данной системе составляет статическая ошибка , время установления составляет .

По ТЗ, перерегулирование должно равняться 0, так же как и статическая ошибка, поэтому в систему необходимо добавить ПИД-регулятор. Блок-схема для настройки данного регулятора показана на рис.3.6.

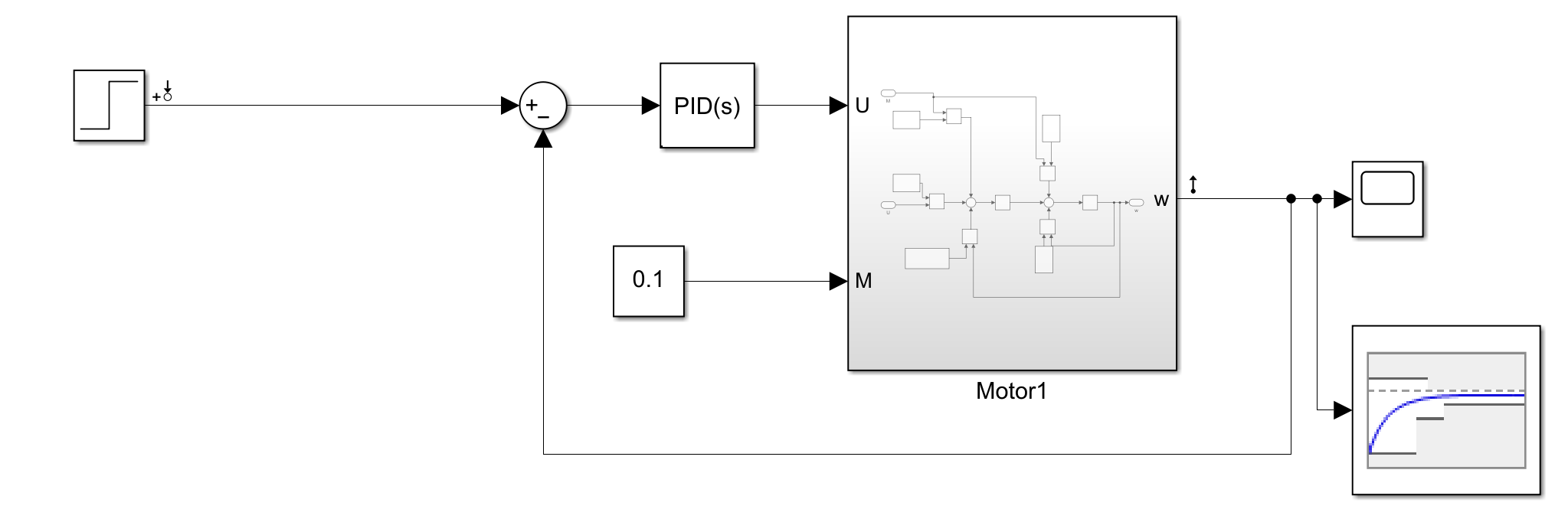


Рис.3.6. Блок-схема настройки ПИД-регулятора

Создав в командном окне MATLAB переменные для хранения коэффициентов регулятора Kp, Ki, Kd, настроим блок Check Step Response

Characteristics на работу с ними (рис.3.7).

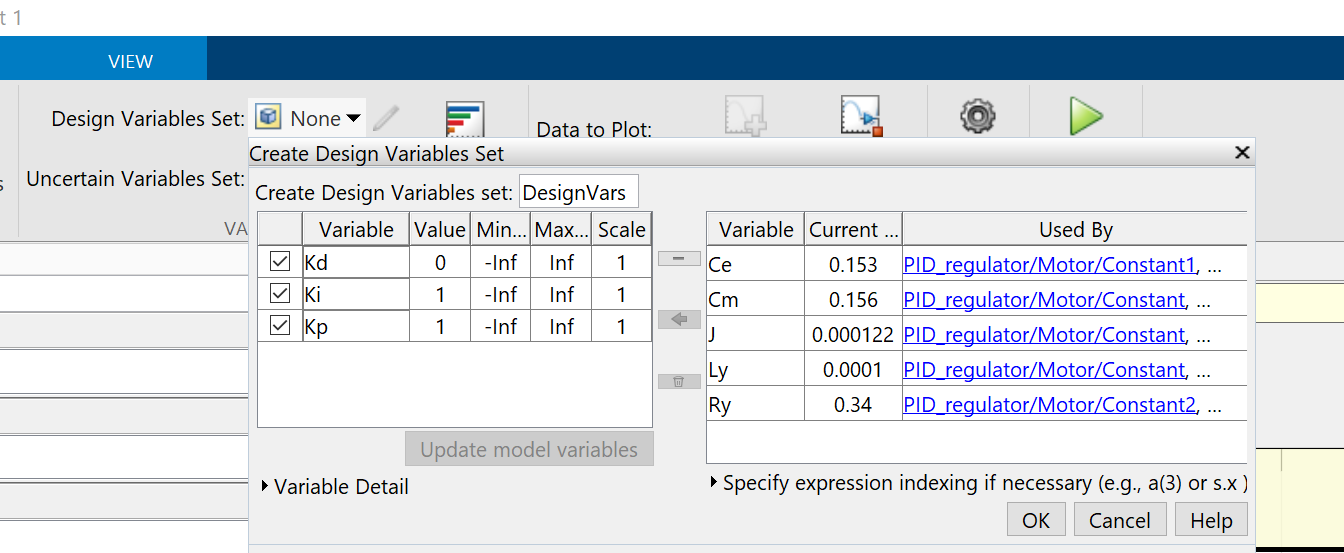


Рис.3.7. Настройки блока Check Step Response Characteristics

После настройки имеем переходную характеристику, показанную на рис.3.8.

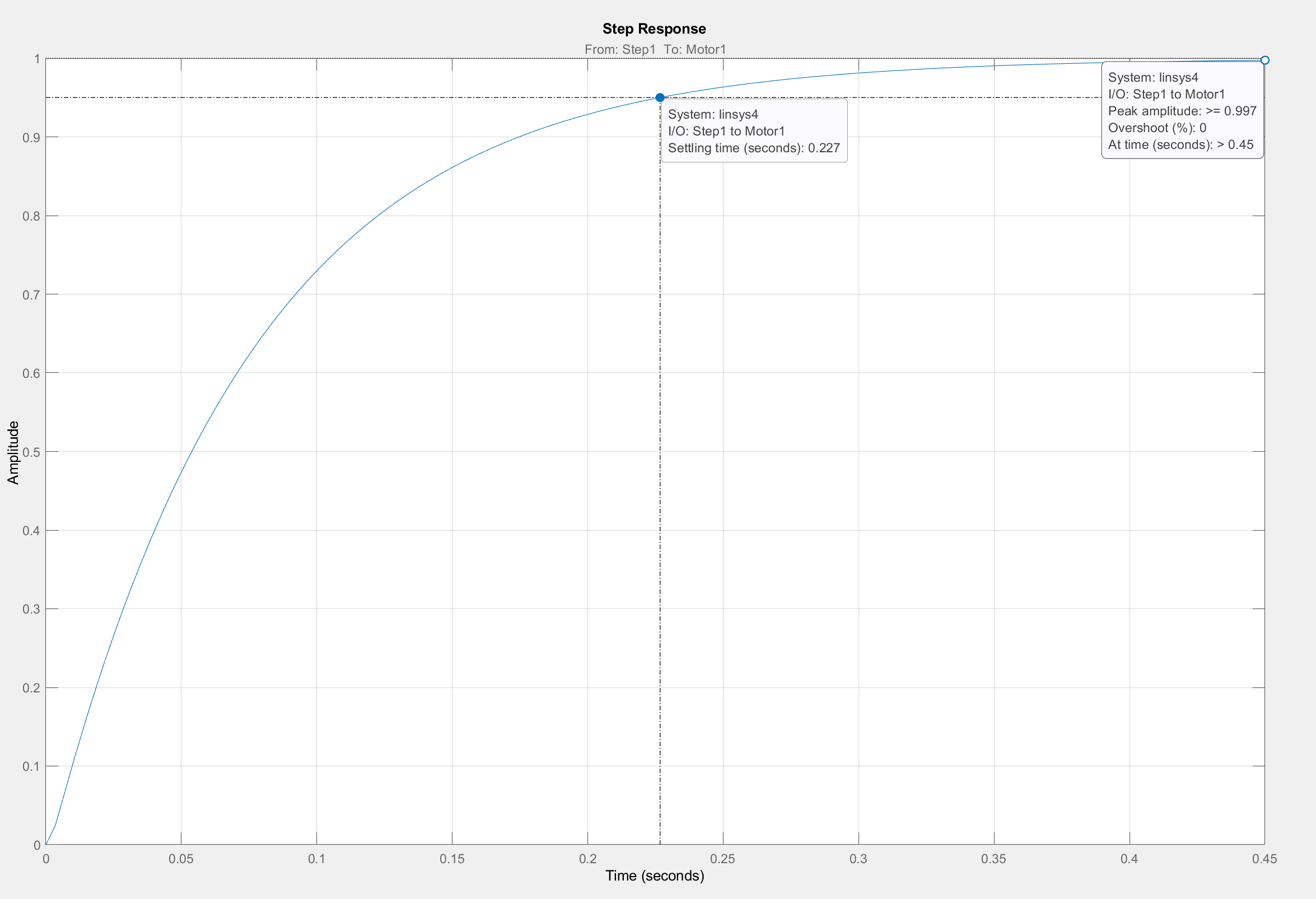


Рис.3.8. Переходная характеристика после настройки

Как видно, перерегулирование статическая ошибка , время установления составляет , что полностью удовлетворяет условию ТЗ. Получившиеся коэффициенты показаны на рис.3.9.

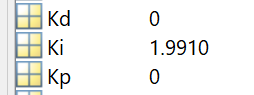


Рис.3.9. Коэффициенты настроенного регулятора

Анализируя полученные коэффициенты, можно сделать вывод о том, что для корректной работы выбранного привода в идеальных условиях достаточно всего лишь И-регулятора, что существенно упростит итоговую систему управления. Но в данной настройке не были учтены все нюансы реальной приводной системы, их мы учтем в следующих главах.

## 3.3.2. Настройка регулятора с учетом нелинейности привода

В прошлом разделе мы настроили САУ привода без учета нелинейности системы мотора. На самом деле в любом приводе присутствуют два нелинейных эффекта – зона нечувствительности и зона насыщения.

Зона нечувствительности *n* (на примере электродвигателя) выражается в том, что электрический двигатель имеет определённый минимальный ток трогания (*i=n*), до достижения которого вал двигателя будет неподвижен. В гидравлическом же двигателе золотник имеет так называемую зону перекрытия (его поршенёк немного шире отверстия, им закрываемого), вследствие чего он откроет путь рабочей жидкости в цилиндр двигателя, только переместившись на некоторую величину *x=n*. Аналогично и в случае пневмодвигателя, где роль золотника играет заслонка.

Зона насыщения обнаруживается в том, что при увеличении тока в приводе управляющего органа сверх некоторого значения *i=b* скорость перемещения управляющего органа остаётся постоянной; также и для гидравлического двигателя, когда окна золотника полностью открыты.

В Simulink за зону нечувствительности отвечает блок Dead Zone, а за зону насыщения – блок Saturation.

Составим блок-схему настройки ПИД-регулятора с учетом эффектов нелинейности, рис.3.10.

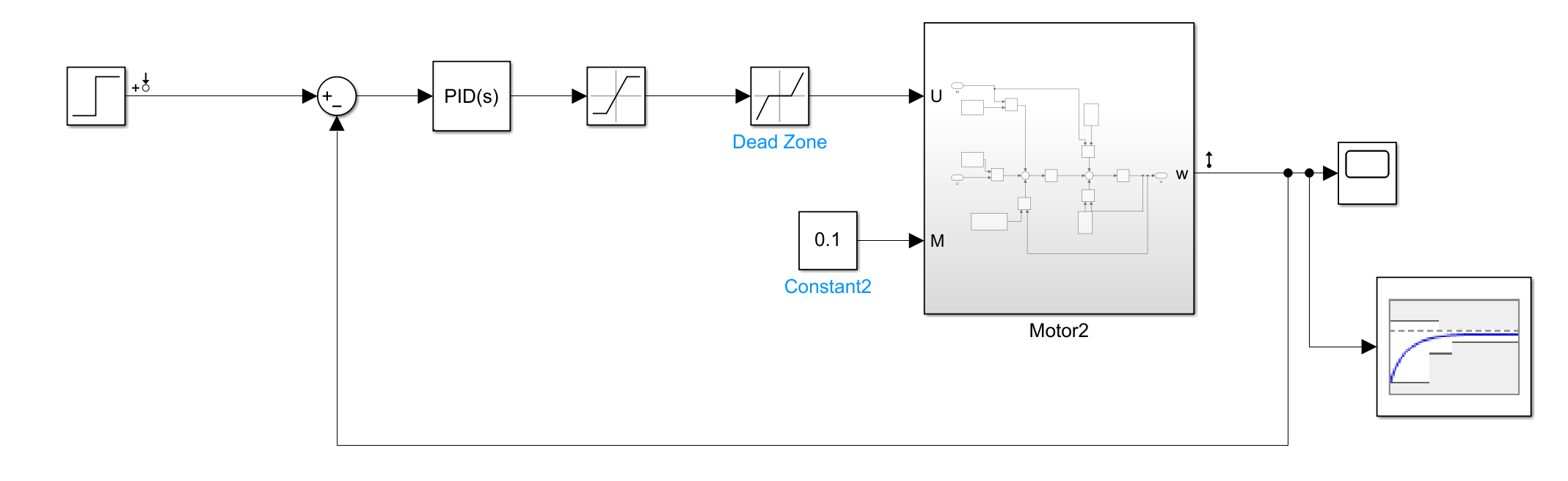


Рис.3.10. Блок-схема настройки регулятора нелинейной САУ

Для Dead Zone значение коэффициента ограничителя возьмем равным а значение коэффициента нечувствительности для Saturation зададим равным

Переходная характеристика нелинейной САУ с закомментированным регулятором показана на рис.3.11.

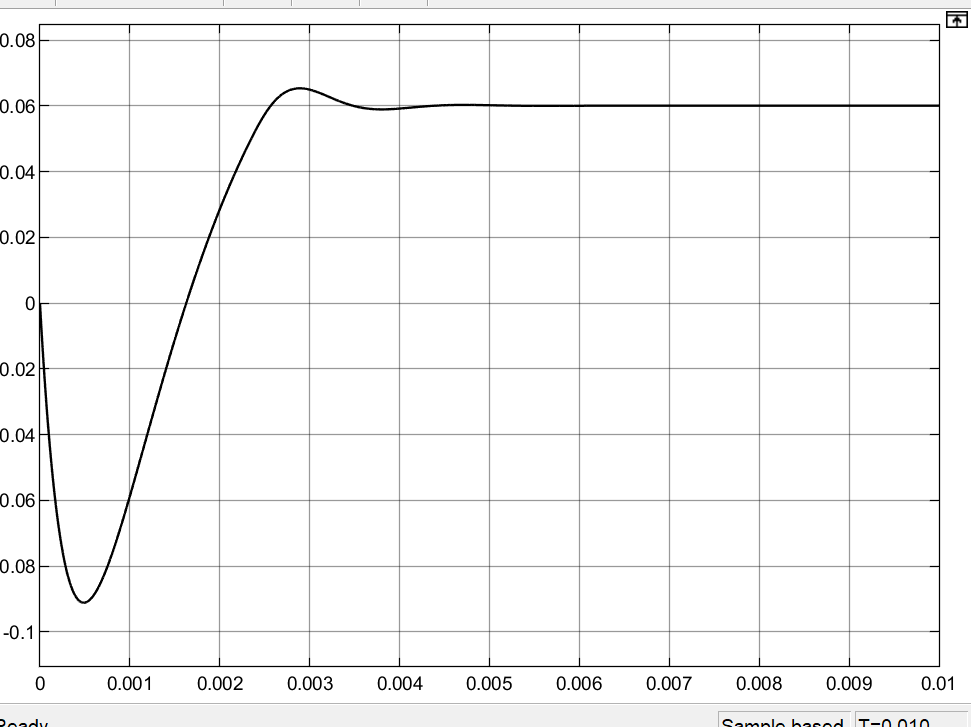


Рис.3.11. Переходная характеристика ненастроенной нелинейной САУ

Как видим, присутствует сильная колебательность и очень большая статическая ошибка , перерегулирование, судя по графику, составляет . Стоит также отметить, что при значении момента сопротивления больше чем 0.006 наблюдается реверс движения двигателя.

Приступим к настройке ПИД регулятора. Для начала настроим переменные для оптимизации, также как это сделано на рис.3.7. Затем проведем оптимизацию. Параметры переходного процесса после оптимизации показаны на рис.3.12.

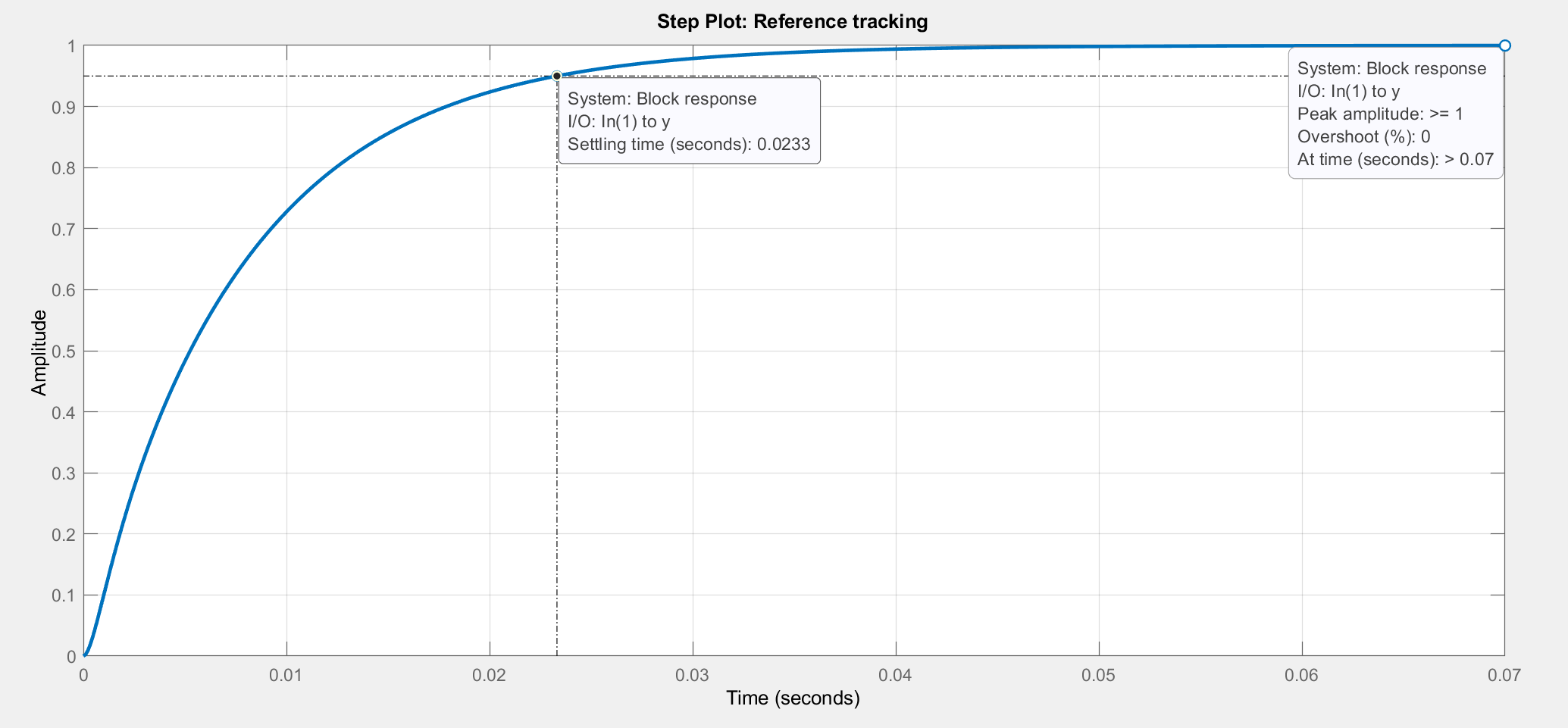
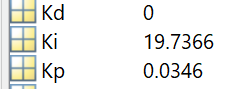


Рис.3.12. Характеристики настроенной САУ

Коэффициенты ПИД-регулятора получились такими:



Таким образом, в данном разделе была настроена САУ привода с учетом нелинейности его системы. И опять мы получаем упрощение системы управления за счет того, что получился ПИ-регулятор а не полноценный ПИД. Но в данной главе мы не учитывали такой эффект системы управления, как задержка обратной связи, ее мы учтем в следующей главе

## 3.3.3 Настройка регулятора САУ с учетом задержки обратной связи

В настоящей системе управления отклик на изменение входного сигнала происходит не мгновенно. Это происходит, в основном, из-за задержек, необходимых на получение данных обратной связи и их обработку.

Настроим САУ привода с учетом запаздывания .

В программном пакете MATLAB/Simulink функцию запаздывания реализует блок Transport Delay. Составим схему для настройки САУ с запаздыванием, рис.3.13.

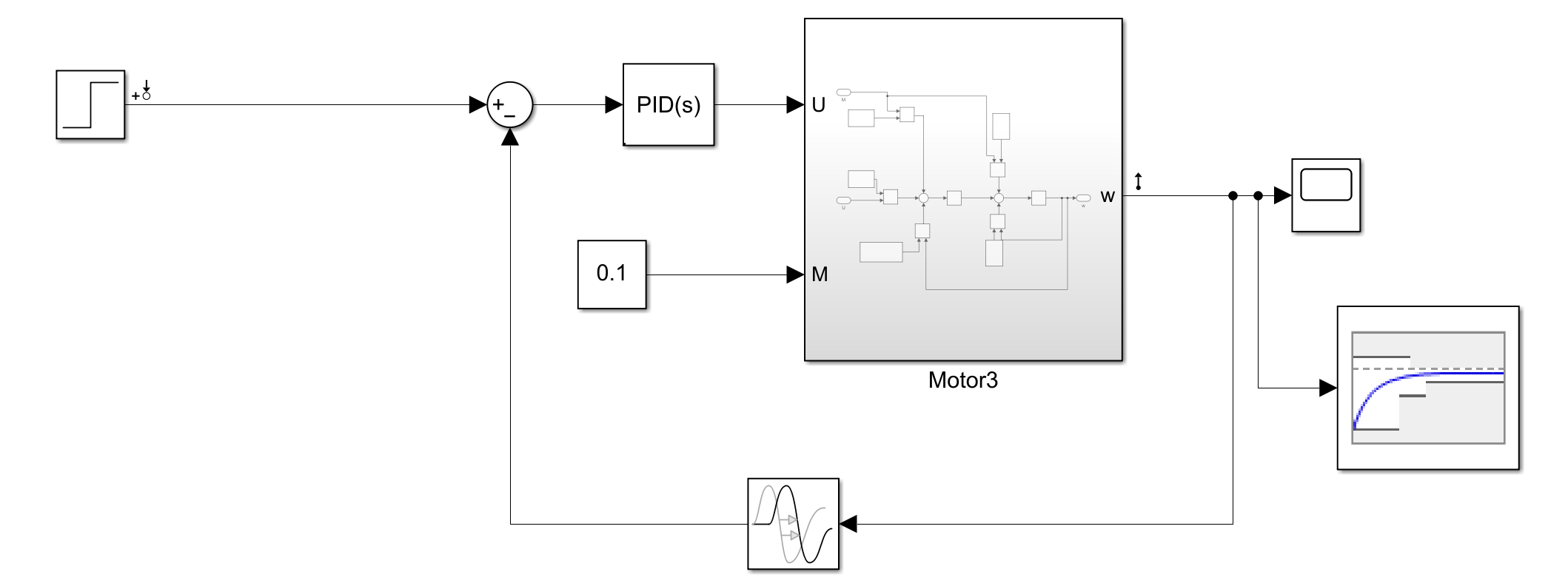


Рис.3.13. Математическая модель САУ с запаздыванием

Узнаем исходные характеристики переходного процесса с помощью Model Linearizer. Полученный график переходного процесса показан на рис.3.14.

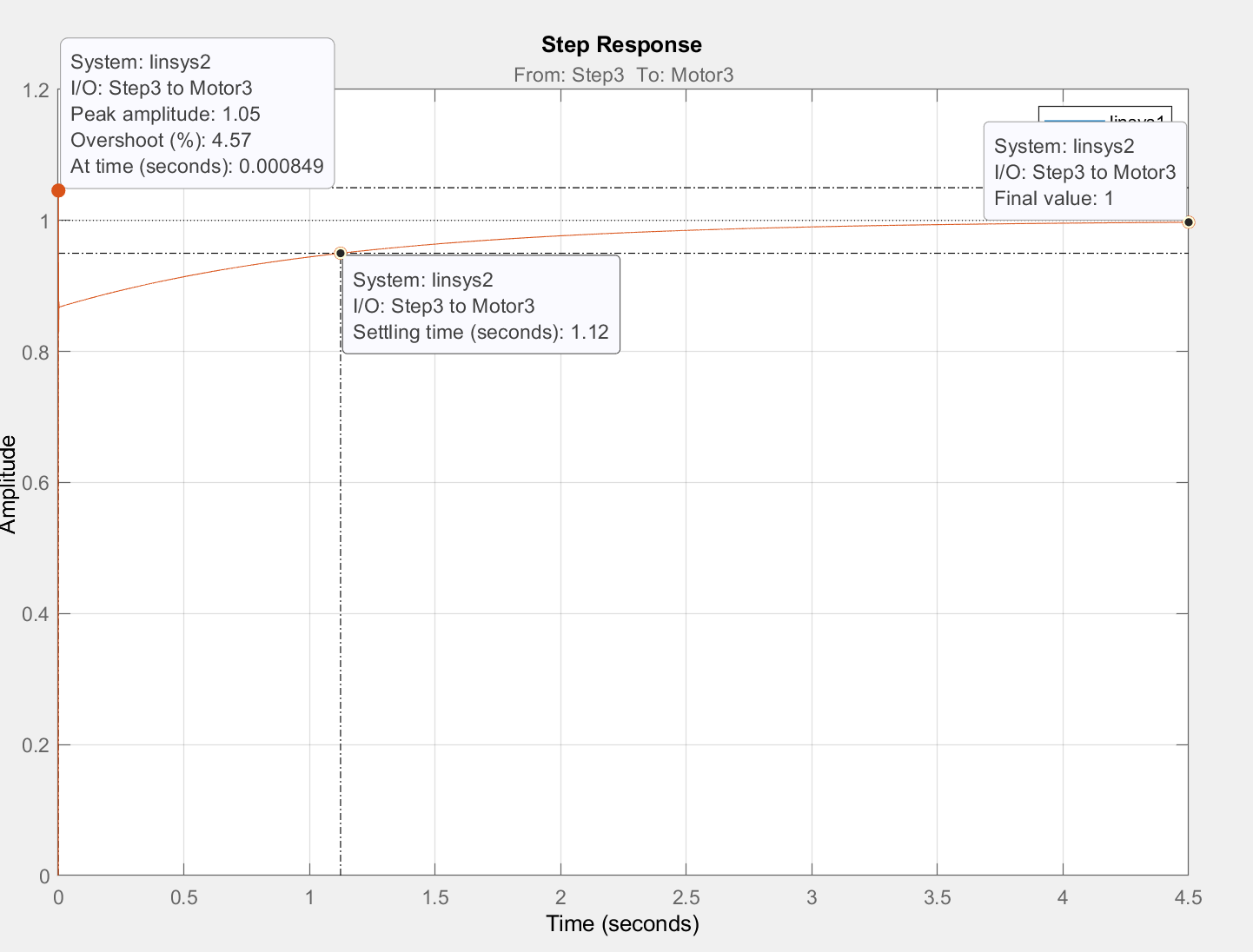


Рис.3.14. Параметры переходного процесса ненастроенной САУ

По графику видим, что перерегулирование , статической ошибки нет, колебательности тоже нет, но форма графика не правильная.

Приступим к настройке ПИД регулятора. Для начала настроим переменные для оптимизации, также как это сделано на рис.3.7. Затем проведем оптимизацию. Параметры переходного процесса после оптимизации показаны на рис.3.15.

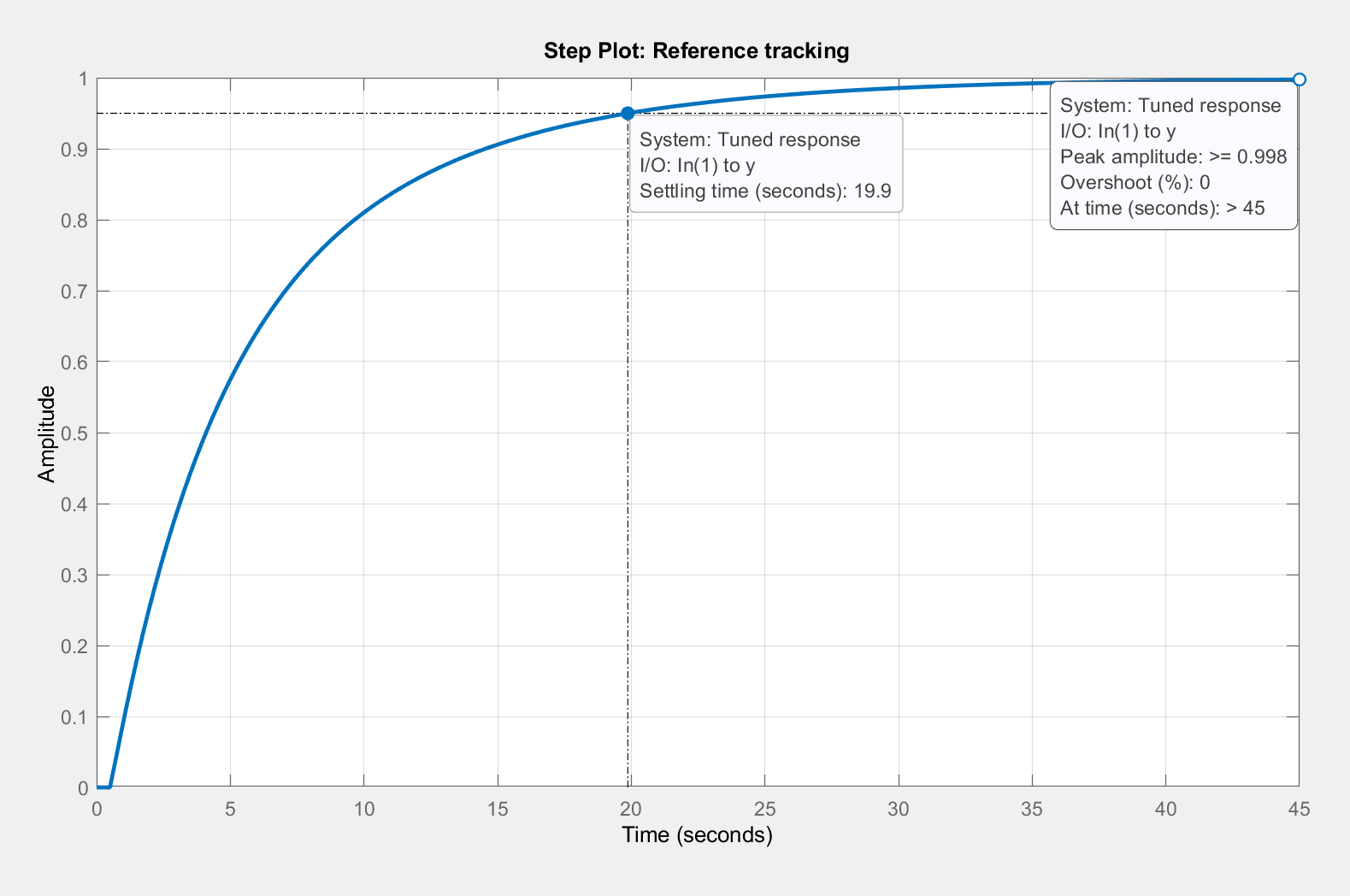
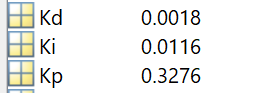


Рис.3.15. Параметры переходного процесса с настроенным регулятором

Как видно, статическая ошибка и перерегулирование были устранены, однако существенно выросло время установления (с 1.12 с до 19 с). Это вызвано как раз наличием задержки в цепи обратной связи. Полученные коэффициенты ПИД-регулятора показаны ниже:



Таким образом в данном разделе мы настроили САУ с учетом наличия задержки 0.5 с в цепи обратной связи. Анализируя полученные во время настройки данные, можно сказать, что задержка на обработку обратной связи в проектируемой САУ не должна быть больше 0.5 секунд, иначе данная система управления не будет справляться с поставленной задачей. В данном случае уже, как видно, нужен полноценный ПИД регулятор. Осталось учесть только один эффект – робастность привода.

## 3.3.4. Настройка робастной САУ

В реальной системе управления параметры двигателя могут колебаться в от двигателя к двигателю. В этом и заключается один из признаков робастной САУ – некоторые их параметры могут изменяться в заданных границах.

Главной задачей синтеза робастных систем управления является поиск закона управления, который сохранял бы выходные переменные системы и сигналы ошибки в заданных допустимых пределах несмотря на наличие неопределенностей в контуре управления.

Попробуем настроить такую систему управления. Блок-схема для проведения настройки показана на рис. 3.16.

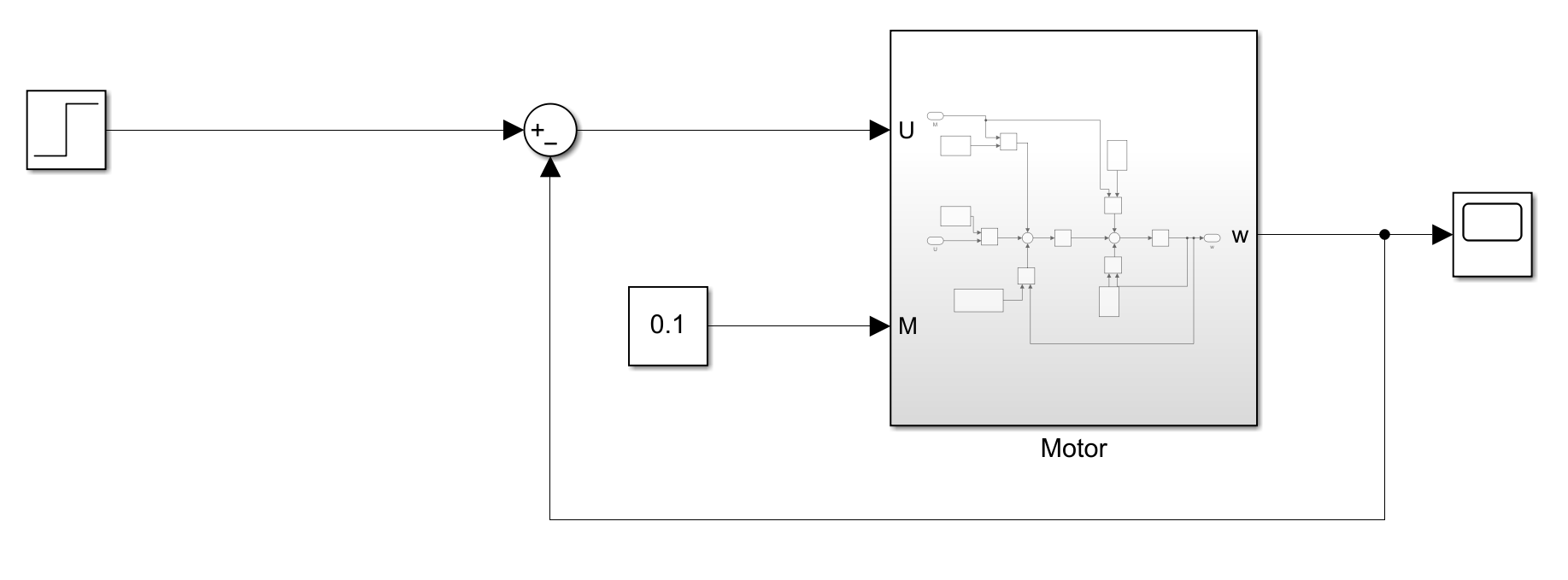


Рис.3.16. Блок-схема настройки робастной САУ

В данном случае изменяемыми являются параметры двигателя (сопротивление якоря () и индуктивность якоря ()). Зададим диапазон изменения этих переменных 20%. Зная это, рассчитаем максимальные и минимальные значения этих параметров:

В меню Uncertain Variables Set блока Check Step Response Characteristics выбираем “неуверенные” переменные, выделим левой кнопкой мыши в правом поле переменные **Ry**, **Ly**, и нажатием стрелки переместим их в левое поле. Нажимаем Set Uncertain Values, где вводим вычисленные значения нижних и верхних границ переменных **Ry** и **Ly** (рис.3.17).

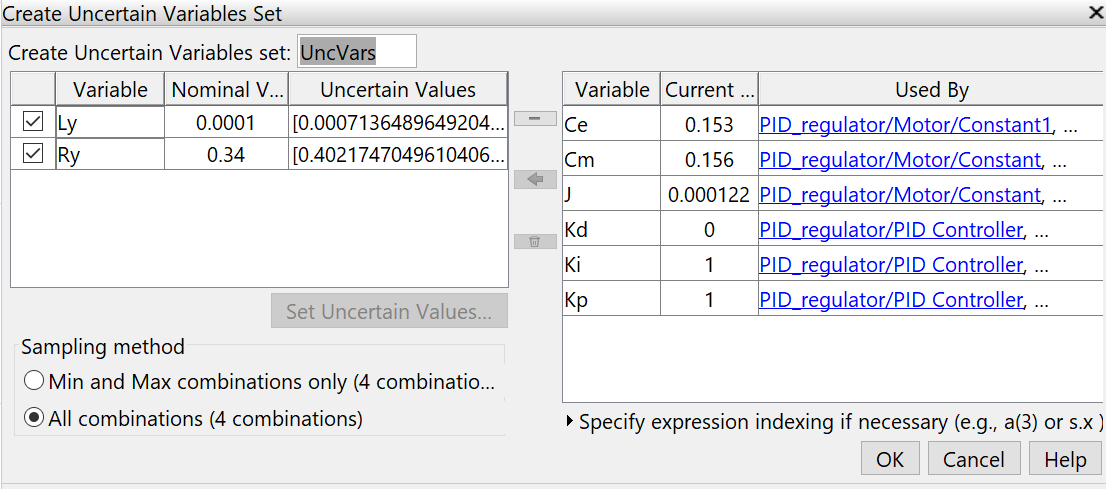


Рис.3.17. Настройка изменяемых параметров регулятора привода

Обычные переменные вводятся в расчёт через меню Design Variable Set.

Начнем процесс оптимизации. После его успешного завершения с помощью

Model Linearizer выведем Step Response для всех комбинаций значений двух ‘не уверенных’ переменных рис.3.18.

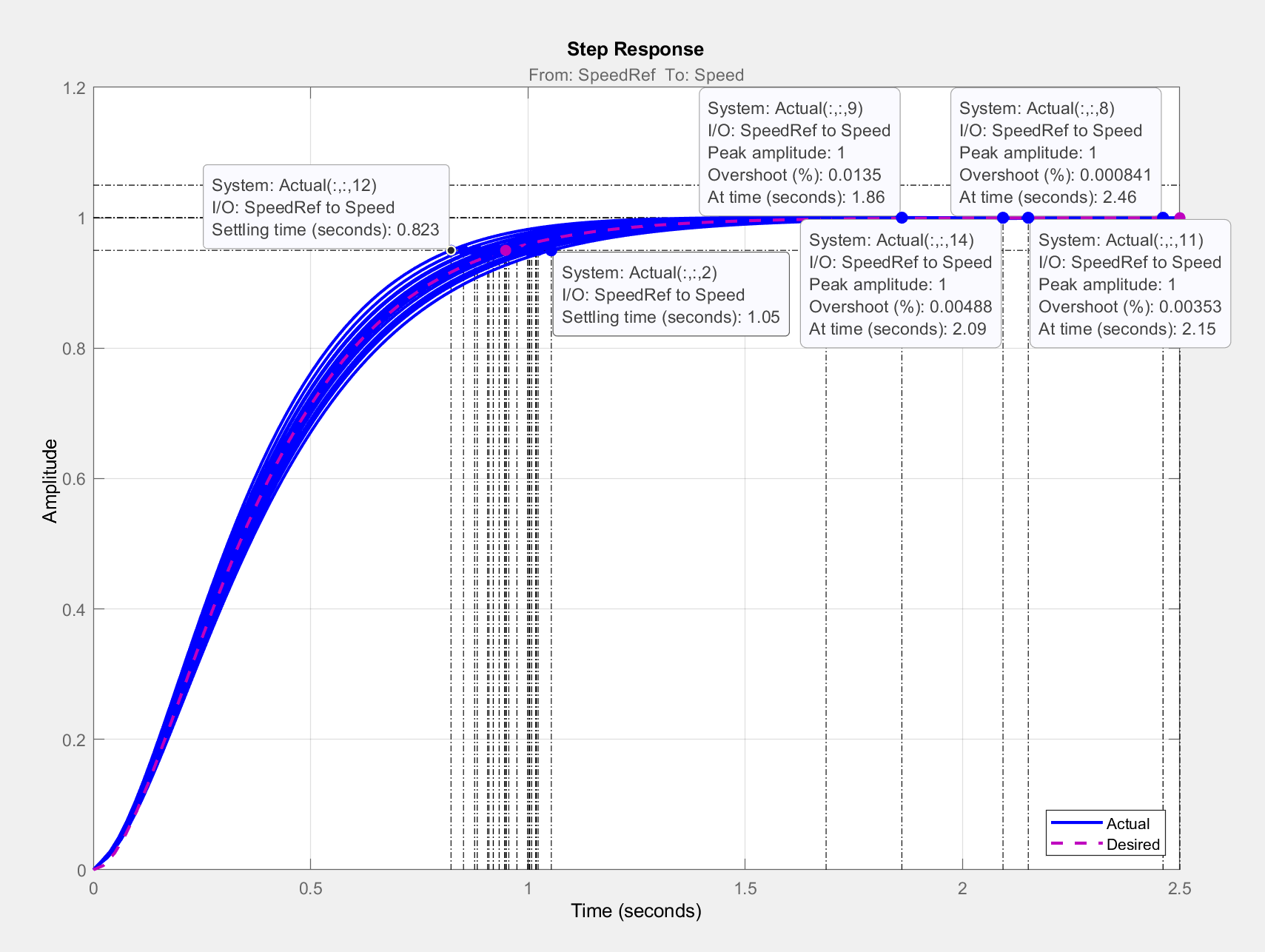


Рис.3.18. Step Response системы при разных комбинациях значений ‘неуверенных’ переменных

Как можно видеть, независимо от комбинации значений переменных, перерегулирование , статическая ошибка , колебательности нет, а время установления находится в промежутке от 0.8 до 1.05с, что полностью удовлетворяет условиям ТЗ.

Коэффициенты ПИД-регулятора получились такими:

Подводя итог все разделу 3.3, можно сказать что в нем были успешно настроены ПИД-регуляторы, обеспечивающие заданные по ТЗ характеристики управления контурами приводов колес, с учетом таких эффектов систем управления привода, как нелинейность, робастность и задержка обработки данных обратной связи.

# 3.4. Проверка устойчивости системы управления

## 3.4.1. Проверка устойчивости системы управления по критерию Гурвица

Критерий устойчивости Гурвица формулируется так: для устойчивости линейной системы необходимо и достаточно, чтобы все n диагональных миноров, полученных из матрицы Гурвица (составленной из коэффициентов характеристического уравнения замкнутой системы), были положительны.

Рассмотрим устойчивость системы с ПИД-регулятором с коэффициентами от настройки робастной САУ (). Передаточная функция ПИД-регулятора выглядит так:

Передаточная функция нашего двигателя выглядит так (выражение 3.6):

Нужно найти передаточную функцию разомкнутой системы как произведение передаточных функций двигателя и ПИД-регулятора:

(3.10)

Вычислим значения коэффициентов полинома знаменателя:

Подставим числовые значения в (3.10):

Теперь чтоб найти матрицу Гурвица, надо найти передаточную функцию замкнутой функцию. Введем переобозначения:

С помощью новых обозначений запишем передаточную функцию замкнутой системы:

(3.11).

Теперь составим из коэффициентов характеристического уравнения полинома знаменателя матрицу Гурвица H (так как высшая степень у нас 3 – матрица будет 3х3):

Найдем теперь определители главный миноров:

Раз все определитель главных миноров больше 0 то по критерию Гурвица система устойчива.

## 3.4.2 Проверка устойчивости системы управления критерием Михайлова

Критерий устойчивости системы управления Михайлова звучит так:

для устойчивости линейной системы необходимо и достаточно, чтобы годограф Михайлова при изменении от 0 до начинался на вещественной оси и проходил последовательно против часовой стрелки *n* квадрантов комплексной плоскости, не обращаясь в нуль и стремясь к в *n*-м квадранте.

Годографом Михайлова называют график на комплексной плоскости полинома знаменателя замкнутой системы.

Возьмем передаточную функцию замкнутой системы, полученную в выражении (3.11):

Теперь возьмем характеристическое уравнение (полином знаменателя):

(3.12)

, и проведем замену оператора Лапласа p на :

(3.13)

Выражение (3.13) является функцией для построения годографа Михайлова. По полученному выражению построим в среде MathCAD характеристику, отображающую состояние исследуемой САУ (рис.3.19).



Рис.3.19. Годограф Михайлова

Как можно видеть на рис.3.19, годограф начинается из положительной точки на вещественной оси, не обращается в нуль и уходит в бесконечность в 3-ем квадранте, что подтверждает устойчивость системы.

## 3.4.3. Проверка устойчивости системы критерием Найквиста

Критерий устойчивости Найквиста формулируется так: для устойчивости замкнутой системы необходимо и достаточно, чтобы амплитудно-фазовая характеристика устойчивой разомкнутой системы при изменении от 0 до не охватывала точку с координатами .

Запишем передаточную функцию замкнутой системы (выражение 3.11):

Найдем обратную ей функцию:

Теперь заменим в ней оператор Лапласа p на :

(3.14)

По выражению (3.14) построим график амплитудно-фазовая характеристика устойчивой разомкнутой системы (рис.3.20).



Рис.3.20. АФЧХ устойчивой разомкнутой системы

Как видно из рис.3.20, АФЧХ устойчивой разомкнутой системы не охватывает точку комплексной плоскости с координатами , значит замкнутая система устойчива.

## 3.4.4. Проверка устойчивости систему управления критерием Боде и определение запаса устойчивости

Критерий устойчивости Боде формируется так: для устойчивости замкнутой системы необходимо и достаточно, чтобы на всех частотах, где ЛАЧХ разомкнутой системы положительная (L() > 0), фазовый сдвиг не достигал значения -180 или достигал его четное число раз.

Для того, чтобы проверить нашу передаточную функцию разомкнутой системы (выражение 3.10, ) по критерию Боде, введем ее с помощью функции tf() в командное окно MATLAB:

%коэффициенты

a2 = 0.001;

a1 = -0.09;

a0 = 0.928;

b3 = 7.8e-8;

b2 = 2.7e-4;

b1 = 0.153;

b0 = 0;

W = tf([a2 a1 a0], [b3 b2 b1 b0]); % передаточная функция

Затем функцией bode(W) построим диаграмму ЛАЧХ по критерию Боде (рис.3.21).

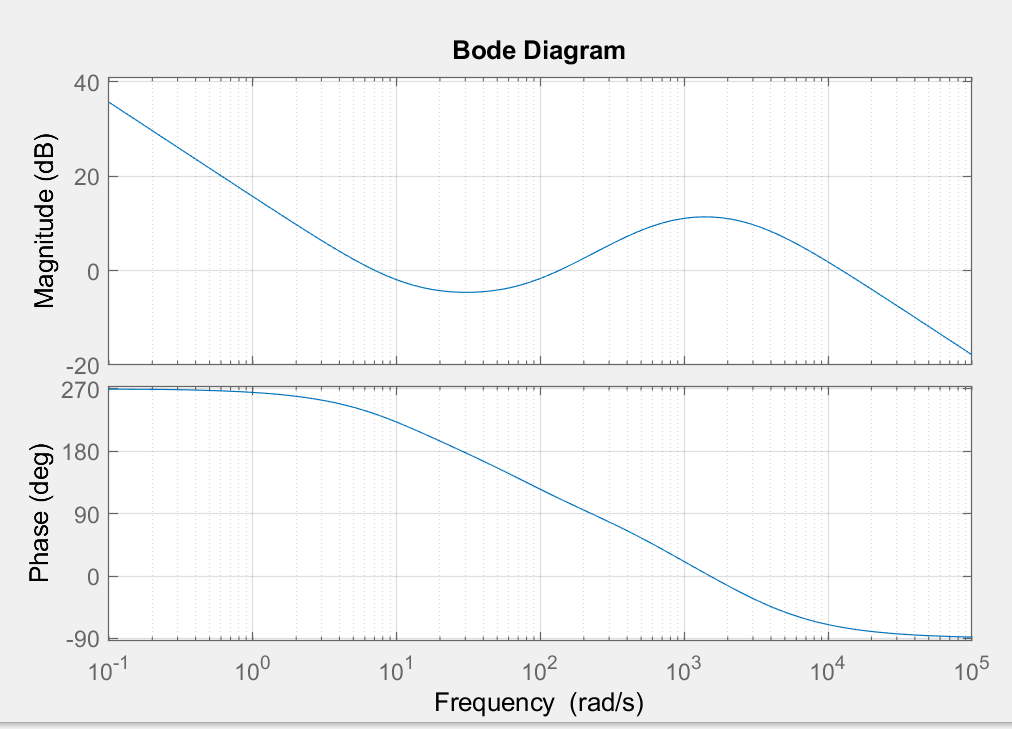


Рис.3.21. Диаграмма Боде разомкнутой системы

Как можно видеть по рис. 3.21, там где ЛАЧХ положительна, фазовый сдвиг не достигает отметки -180 , что доказывает устойчивость данной системы.

Последним действием c помощью команды margin(W) выведем диаграмму Боде с отмеченными на ней запасами устойчивости исследуемой системы (рис.3.22.).

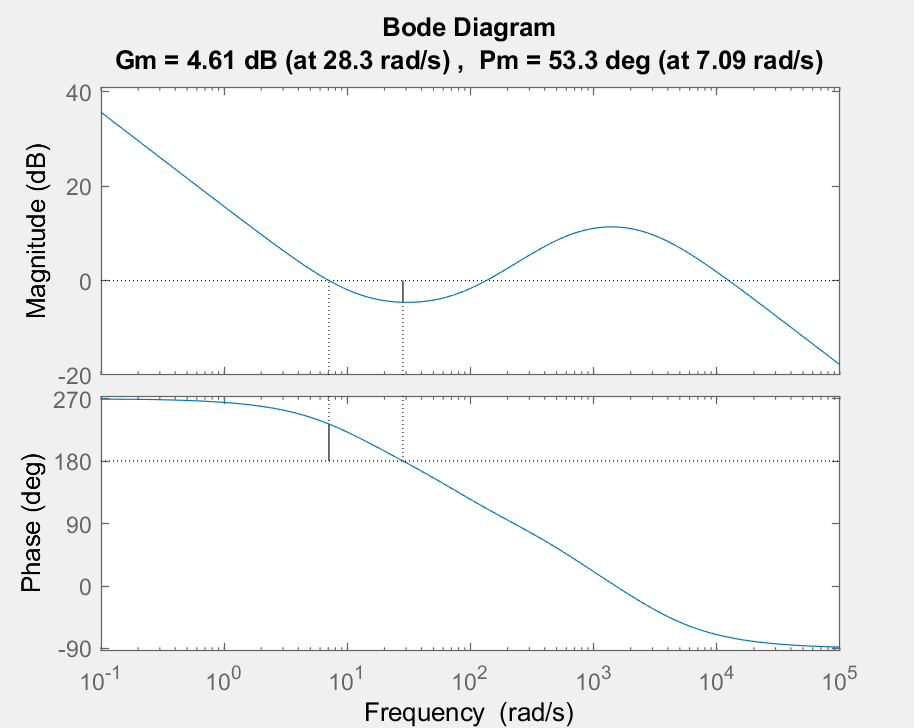


Рис.3.22. Диаграмма Боде с отмеченным запасом устойчивости

Как видно из рис.3.22, запас по амплитуде (Gm) равен 4.6 дБ, что значит можно ставить коэффициент усиления в пределах этого значения и система останется стабильна, а запас по фазе (Pm) составляет 53, что позволит сдвигать фазу системы в пределах этого значения также без потери устойчивости.

Таким образом, в данном разделе мы подтвердили устойчивость системы управления с робастным ПИД-регулятором методами Гурвица, Михайлова, Найквиста и Боде, а также узнали запасы устойчивости по амплитуде и по фазе для данной системы.

# 3.5. Настройка нечеткого регулятора

Нечёткий регулятор (англ. *fuzzy controller*) — регулятор, построенный на базе нечеткой логики.

Для реализации нечеткого регулятора необходимо:

1. Определить входные лингвистические переменные. Например, «Ошибка» и «Изменение ошибки».
2. Определить лингвистическую переменную, которую мы хотим получить. В данном случае это будет лингвистическая переменная «Напряжение».
3. Определить правила образования результирующей переменной из входных переменных.

В результате, мы получим возможность определять необходимые нам значения, такие как напряжение, подаваемое на двигатель или любое другое необходимое значение.

Блок-схема для настройки данного регулятора показана на рис.3.23.

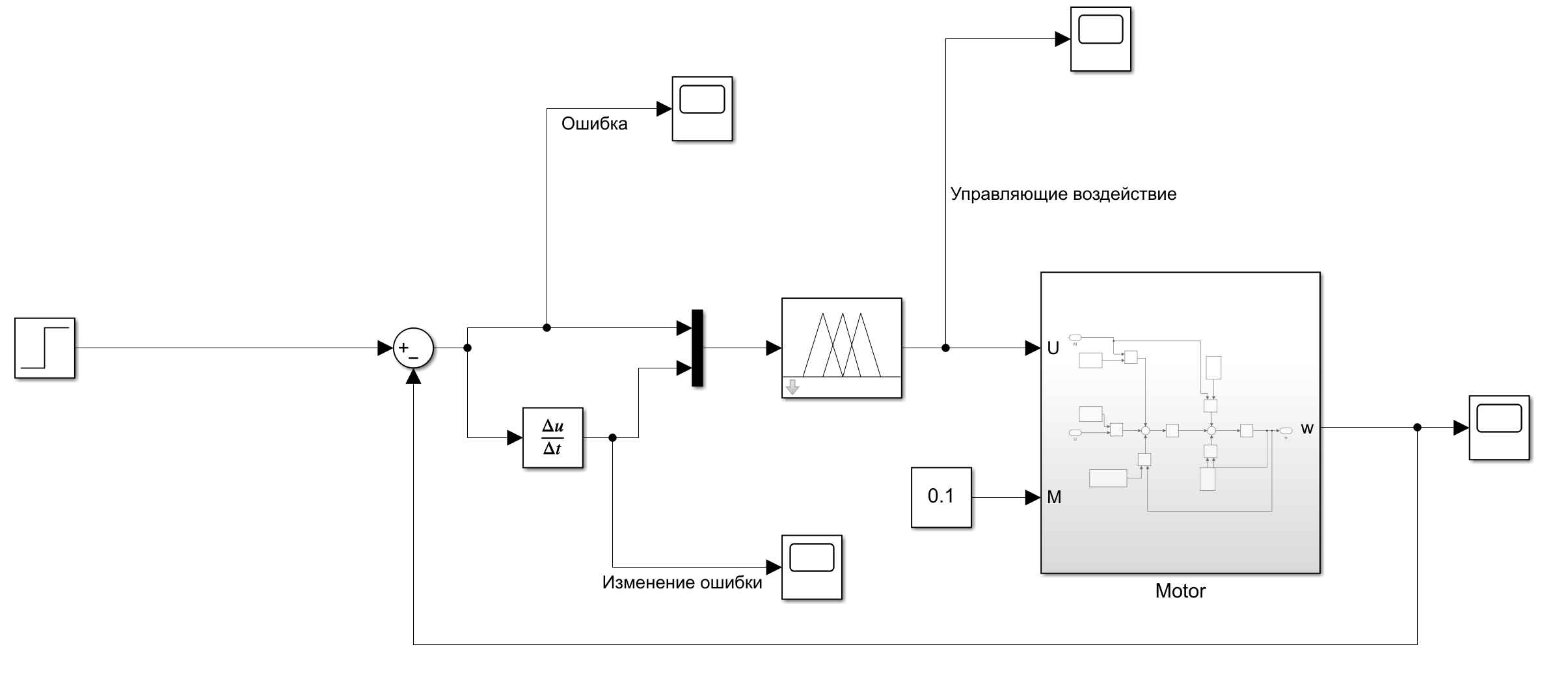


Рис.3.23. Блок-схема настройки нечеткого регулятора

На вход блока нечеткого регулятора через мультиплексор подаются два сигнала: сигнал расхождения текущего значения угловой скорости и желаемым ее значением (сигнал ошибки), а также сигнал изменения ошибки изменения во времени (сигнал скорости изменения ошибки), выходом же данному блоку служит одно лишь значение управляющего напряжения.

По количеству входных и выходных сигналов с помощью интерфейса настройки нечетких регуляторов настроим входные и выходные лингвистические переменные (ошибка и скорость изменения ошибки – входные лингвистические переменные, напряжение – выходная лингвистическая переменная) рис.3.24.

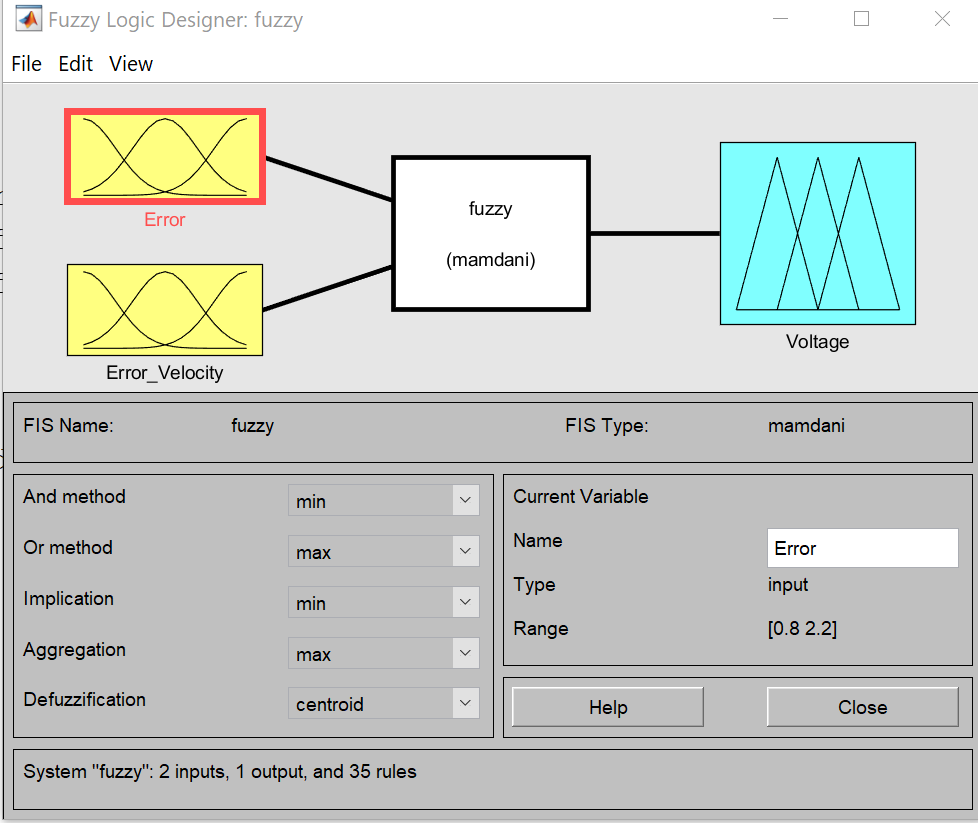


Рис.3.24. Интерфейс настройки лингвистических переменных

Путем предварительных оценок работы системы и того, в каких диапазонах ошибок и скоростей их изменения она будет работать, были определены термы для каждой лингвистической переменной (рис. 3.26, 3.25 и 3.27). Настройка термов осуществляется путём указания диапазонов для графиков функций принадлежности (графа **Params**); там же можно выбрать тип графика функции принадлежности (графа **Type**). Таким образом была выполнена процедура *фаззификации*.

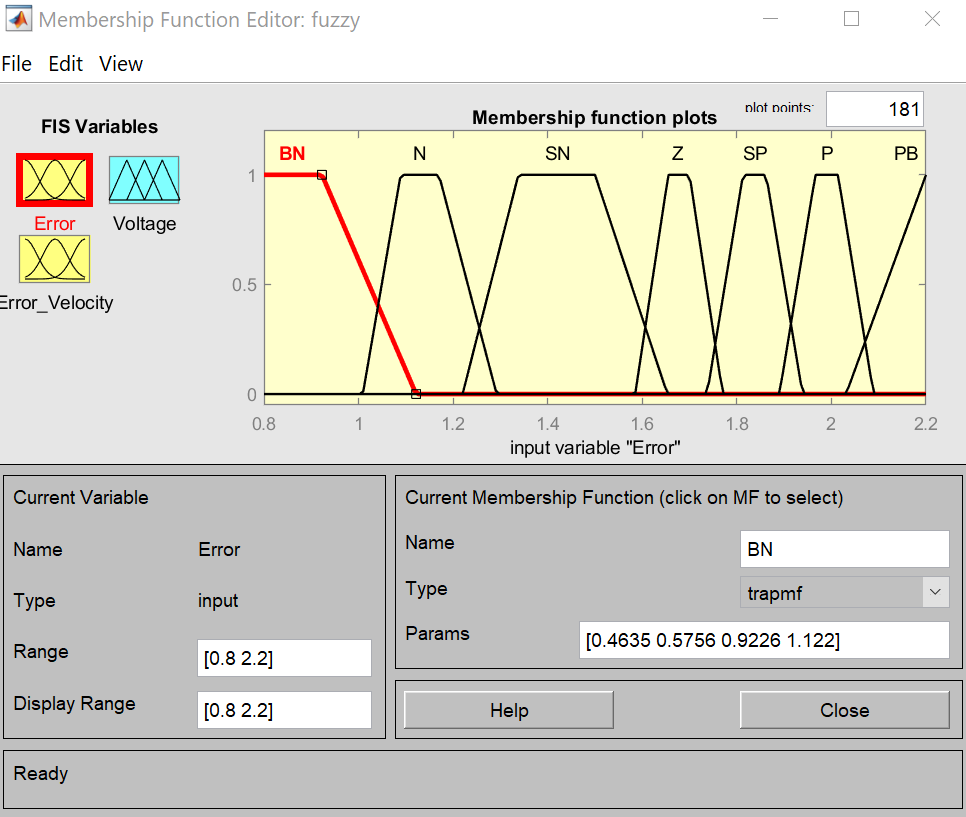


Рис.3.26. Настройки терм для входной лингвистической переменной ‘Ошибка’

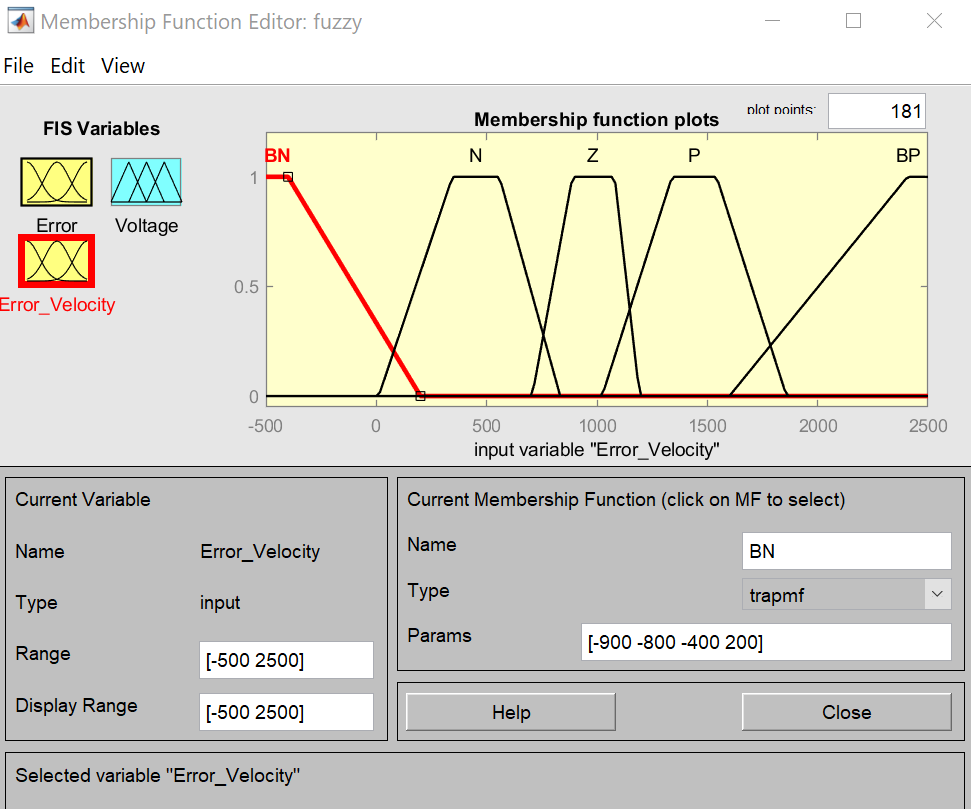


Рис.3.27. Настройки терм для входной лингвистической переменной ‘Скорость изменения ошибки’

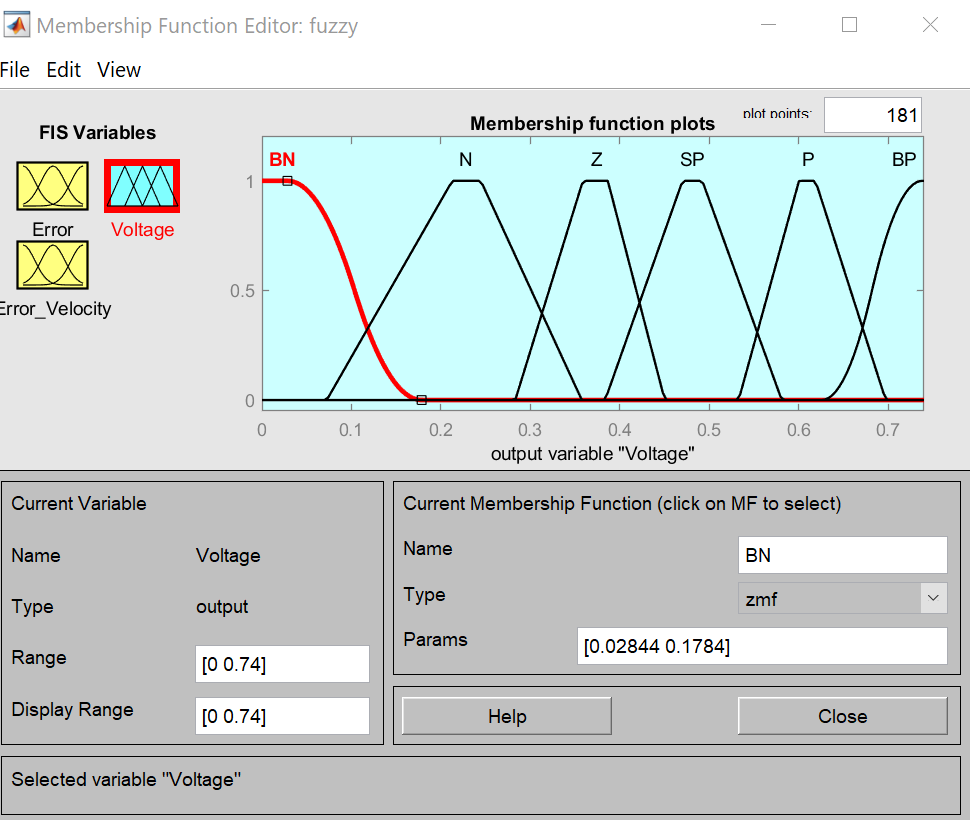


Рис.3.28. Настройки терм для выходной лингвистической переменной ‘Напряжение’

Также были заданы правила нечеткого регулятора (рис.3.29). Правила формируются следующим образом: в трёх графах (**error is, error\_velocity is, Voltage is**) выбираются необходимые термы лингвистических переменных, после чего, нажимая на кнопку **Add rule,** мы создадим правило для нечёткого регулятора. Всего было сформировано 35 правил. Таким образом была выполнена процедура *дефаззификация*.

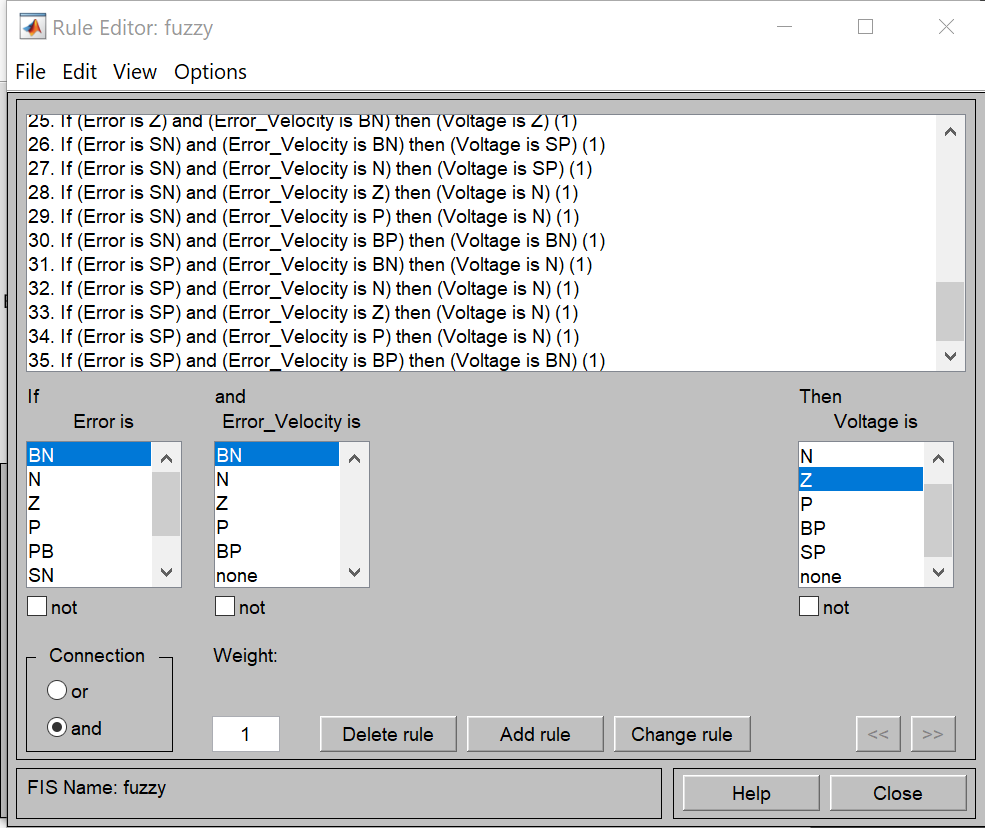


Рис.3.29. Список правил нечеткого регулятора

После экспорта настроенного fuzzy-регулятора в Matlab-переменную, построим график переходного процесса (рис.3.30).

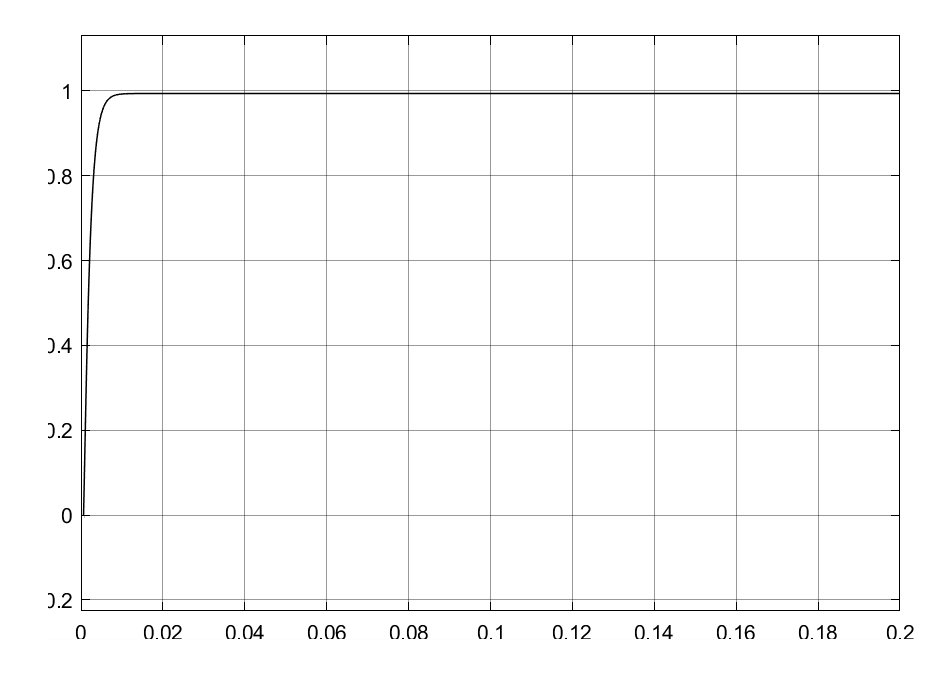


Рис.3.30. График переходного процесса с настроенным нечетким регулятором

Как можно видеть по рис.3.30,

Далее, построим графики ошибки, изменения ошибки и управляющего воздействия (рис.3.30, 3.31, 3.32).

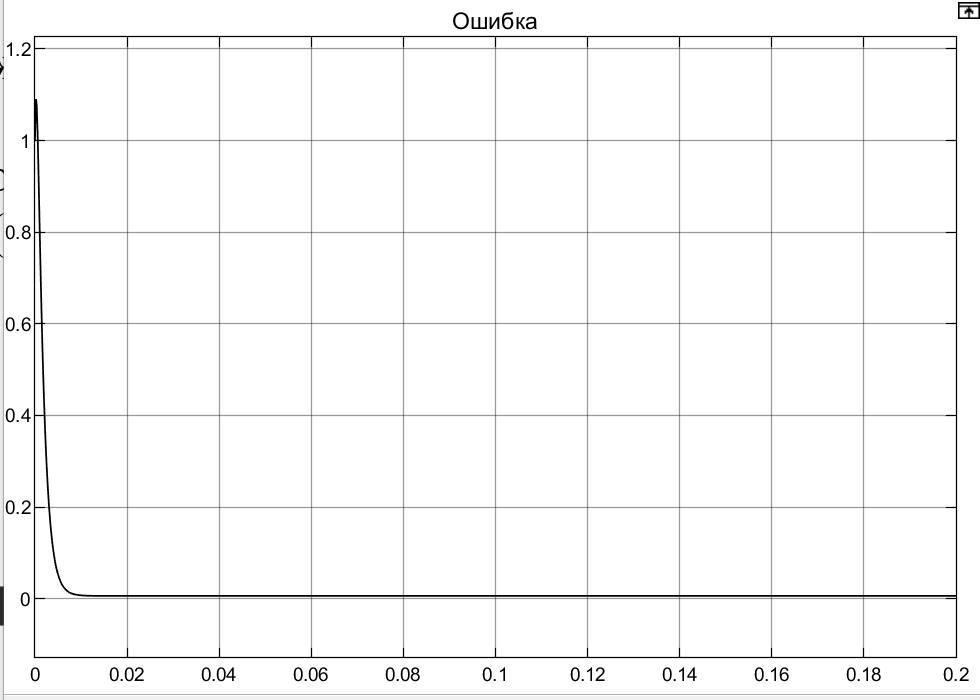


Рис.3.30. График ошибки переходного процесса с настроенным нечетким регулятором

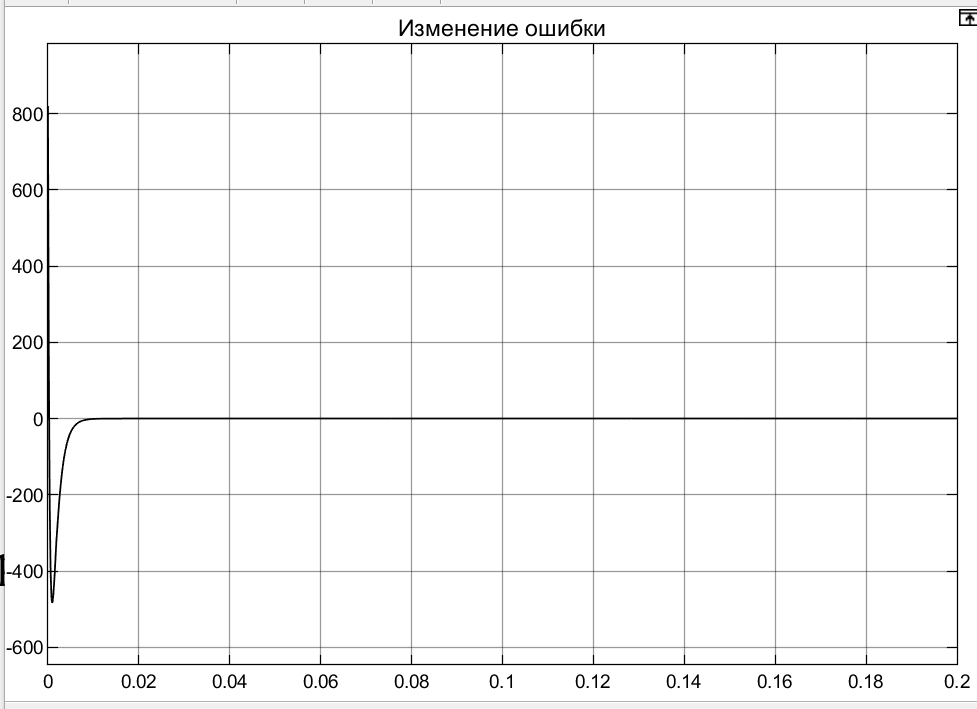


Рис.3.31. График ошибки переходного процесса с настроенным нечетким регулятором

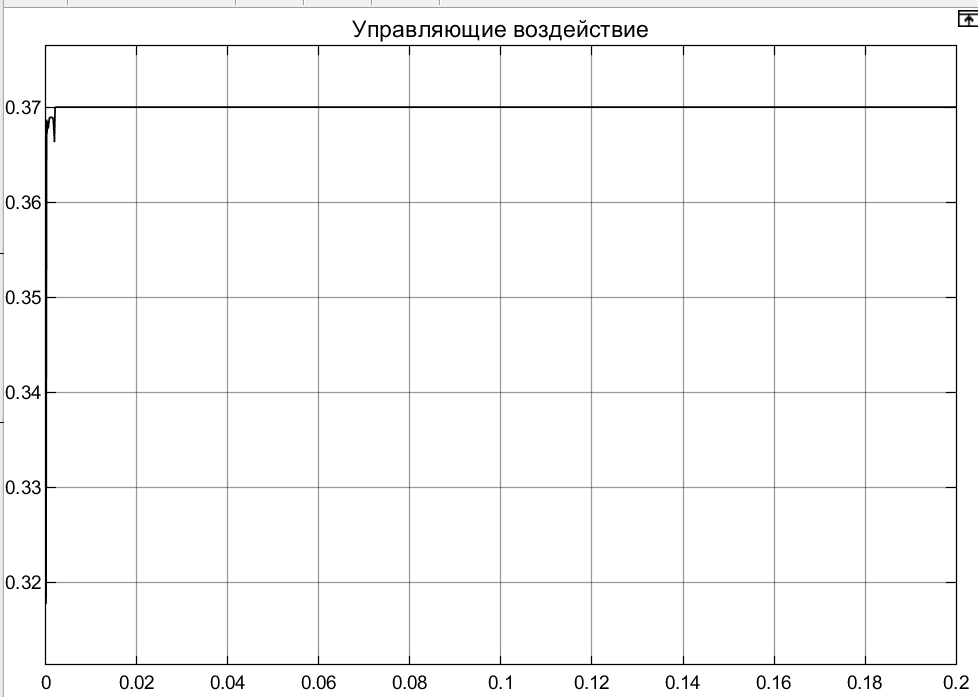


Рис.3.31. График управляющего воздействия нечеткого регулятора

Анализируя графики, мы можем видеть, что выходное управляющее воздействие нечёткого регулятора является многоступенчатым, что говорит о том, что при его формировании задействовано большинство из составленных нами правил. 3-D изображение зависимости трех лингвистических переменных представлено на рисунке 3.32.

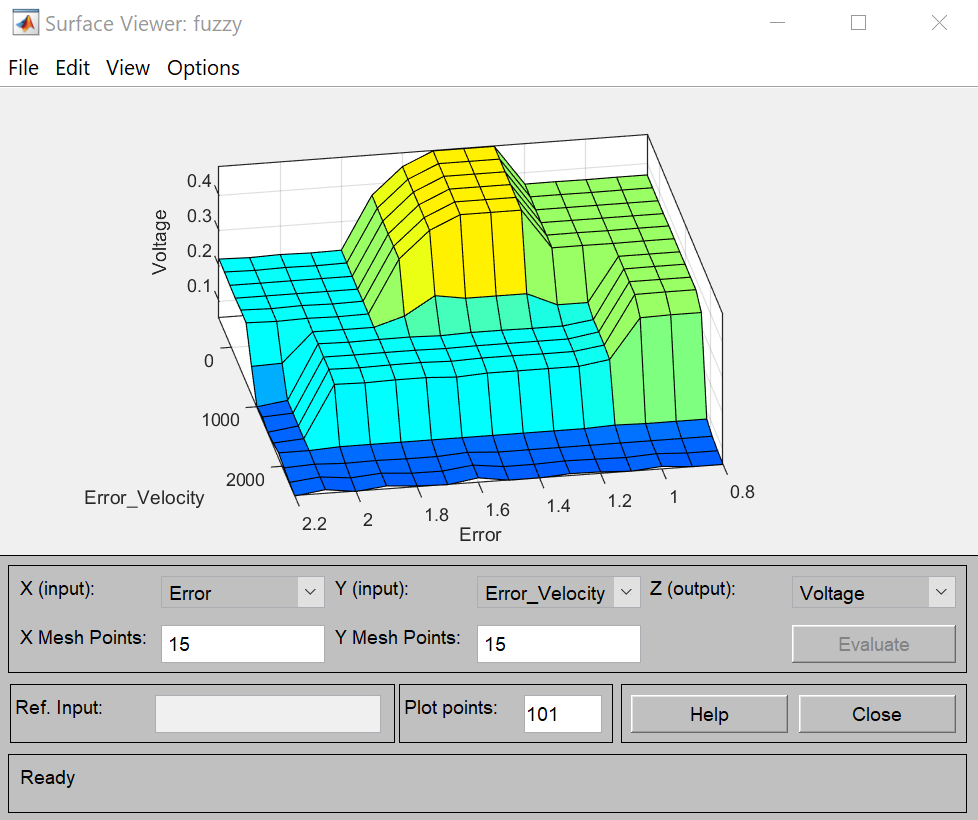


Рис.3.32. 3-D изображение зависимости трех лингвистических переменных

По данному графику можно судить о правильности настройки регулятора: на поверхности не должно быть резких спадов и скачков, а значения переменной **Voltage** должны плавно возрастать от 0 до заданного максимума.

Полученное качество переходного процесса ( ) удовлетворяет требованиям, следовательно, настройку нечеткого регулятора можно считать успешной.

Таким образом, в данной главе был успешно настроен нечеткий регулятор системы управления приводом колеса. Сравнивая качество переходного процесса с ПИД-регулятором и нечетким регулятором, видно, что нечеткий регулятор намного быстрее справляется с задачей, время установления системы с нечетким регулятором более чем в 100 раз меньше времени установления той же системы с ПИД-регулятором.