



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(национальный исследовательский университет)»

Институт №3

Кафедра №301

Курсовой проект по дисциплине

«Проектирование систем автоматического управления ЛА»

на тему:

**«Проектирование системы автоматического управления
боковым движением самолёта на режиме полета №1»**

Выполнил студент группы М30-501С-18
Дзущев С. С.

Принял преподаватель:
Мулин П. В.

Москва 2022

Содержание

Задание.....	3
Исходные данные и требования к системам	4
1. Математические модели бокового движения самолёта.....	5
1.1. Уравнения бокового движения самолёта.....	5
1.2. Упрощённые модели бокового движения	8
2. Система стабилизации угла крена	11
2.1. Теоретические сведения	11
2.2. Синтез системы управления углом крена.....	13
2.3. Подбор коэффициентов в NCD-блоке для САУ углом крена	17
2.4. Построение области устойчивости для САУ углом крена	19
2.5. Построение области устойчивости при использовании достаточных условий устойчивости для САУ углом крена и сравнение такого метода с методом построения по критерию Рауса.....	20
3. Система стабилизации угла рысканья в режиме плоского разворота	22
3.1. Теоретические сведения	22
3.2. Синтез системы управления углом рыскания в режиме плоского разворота	24
3.3. Подбор коэффициентов в NCD-блоке для САУ углом рыскания в режиме плоского разворота	26
3.4. Построение области устойчивости для САУ углом рыскания в режиме плоского разворота	27
3.5. Построение области устойчивости при использовании достаточных условий устойчивости для САУ рыскания при плоском развороте и сравнение такого метода с методом построения по критерию Рауса.....	28
4. Система стабилизации угла рыскания при использовании координированного разворота	30
4.1. Теоретические сведения	30
4.2. Синтез системы управления углом рыскания в режиме плоского разворота	34
4.3. Подбор коэффициентов в NCD-блоке для САУ углом рыскания в режиме координированного разворота	36
4.4. Построение области устойчивости для САУ углом рыскания в режиме координированного разворота	37
Заключение	44
Список использованных источников	45
Приложение 1	46
Приложение 2.....	52
Приложение 3.....	60

Задание

1. Разработать систему управления углом крена на указанном режиме полета, удовлетворяющую требованиям к точности, быстродействию и качеству процессов управления.

- Разработать функциональную и структурную схемы.
- Выбрать параметры законов управления.

2. Разработать систему управления углом рыскания при использовании плоского разворота на указанном режиме полета, удовлетворяющую требованиям к точности, быстродействию и качеству процессов управления.

- Разработать функциональную и структурную схемы.
- Выбрать параметры законов управления.

3. Разработать систему управления углом рыскания при использовании координированного разворота на указанном режиме полета, удовлетворяющую требованиям к точности, быстродействию и качеству процессов управления:

- Разработать функциональную и структурную схемы.
- Выбрать параметры законов управления.

Исходные данные и требования к системам

Линейная математическая модель привода имеет передаточную функцию:

$$W_{\text{пр}}(s) = \frac{\omega_{\text{пр}}^2}{s^2 + 2\xi_{\text{пр}}\omega_{\text{пр}}s + \omega_{\text{пр}}^2}$$

где $\omega_{\text{пр}} = 20 \frac{\text{рад}}{\text{с}}$; $\xi_{\text{пр}} = 0.707$.

N	$H, \text{ км}$	M	$V, \text{ м/с}$	$\bar{Z}^{\beta}, 1/\text{с}$	$\sin \alpha_0$	$\cos \alpha_0$	$\frac{g}{V}, 1/\text{с}$
16	5	0.6	192	-0.2	0.08	1	0.051

$\bar{M}_x^{\beta}, \frac{1}{\text{с}^2}$	$\bar{M}_x^{\omega_x}, \frac{1}{\text{с}}$	$\bar{M}_x^{\omega_y}, \frac{1}{\text{с}}$	$\bar{M}_x^{\delta_{\varepsilon}}, \frac{1}{\text{с}^2}$	$\bar{M}_y^{\beta}, \frac{1}{\text{с}^2}$	$\bar{M}_y^{\omega_x}, \frac{1}{\text{с}}$	$\bar{M}_y^{\omega_y}, \frac{1}{\text{с}}$	$\bar{M}_y^{\delta_{\text{н}}}, \frac{1}{\text{с}^2}$
-5.8	-1.0	-0.2	-7	-3.0	-0.05	-0.2	-2.5

– Переходный процесс отработки заданного угла крена должен удовлетворять требованиям:

$$\sigma \leq 5 \%$$

$t_{p_{\gamma}}$ – минимально

– Переходный процесс отработки заданного угла рыскания при использовании плоского разворота должен удовлетворять требованиям:

$$\sigma = 0 \%$$

$t_{p_{\psi}}^{\text{пл}}$ – минимально

– Переходный процесс отработки заданного угла рыскания при использовании координированного разворота должен удовлетворять требованиям:

$$\sigma = 0 \%$$

$t_{p_{\psi}}^{\text{коорд}}$ – минимально

1. Математические модели бокового движения самолёта

1.1. Уравнения бокового движения самолёта

Дифференциальные уравнения бокового движения самолёта в горизонтальном полёте без крена и скольжения в спокойной атмосфере, записанные в проекциях на оси связанной системы координат, совпадающей с главными осями инерции, имеют вид:

$$\begin{cases} \frac{d\beta}{dt} = \bar{Z}^{\beta} \beta + \sin \alpha_0 \omega_x + \cos \alpha_0 \omega_y + \frac{g}{V} \cos \vartheta_0 \gamma + \bar{Z}^{\delta_n} \delta_n \\ \frac{d\omega_x}{dt} = \bar{M}_x^{\beta} \beta + \bar{M}_x^{\omega_x} \omega_x + \bar{M}_x^{\omega_y} \omega_y + \bar{M}_x^{\delta_{\varepsilon}} \delta_{\varepsilon} + \bar{M}_x^{\delta_n} \delta_n \\ \frac{d\omega_y}{dt} = \bar{M}_y^{\beta} \beta + \bar{M}_y^{\omega_x} \omega_x + \bar{M}_y^{\omega_y} \omega_y + \bar{M}_y^{\delta_{\varepsilon}} \delta_{\varepsilon} + \bar{M}_y^{\delta_n} \delta_n \\ \frac{d\gamma}{dt} = \omega_x - tg \vartheta_0 \omega_y \\ \frac{d\Psi}{dt} = \frac{\omega_y}{\cos \vartheta_0} \end{cases} \quad (1.1.1)$$

где: $\bar{Z} = \frac{Z}{mV}$, $\bar{M}_x = \frac{M_x}{J_x}$, $\bar{M}_y = \frac{M_y}{J_y}$, остальные обозначения – стандартные.

Верхний индекс в коэффициентах уравнений (1.1.1) означает частное дифференцирование по данной переменной, измеренной в радианах в секунду (для угловых скоростей).

При решении различных задач система (1.1.1) обычно упрощается, причём характер упрощений зависит от специфики задачи. Самое распространённое упрощение, состоящее в пренебрежении слагаемыми, величина которых обычно мала, в частности слагаемыми: $\bar{Z}^{\delta_n} \delta_n$, $\bar{M}_x^{\delta_n} \delta_n$, $\bar{M}_y^{\delta_{\varepsilon}} \delta_{\varepsilon}$, $tg \vartheta_0 \omega_y$ кроме того, принимается: $\cos \alpha_0 = \cos \vartheta_0 = 1$.

В результате получается следующая система уравнений бокового движения самолёта:

$$\begin{cases} \frac{d\beta}{dt} = \bar{Z}^\beta \beta + \sin \alpha_0 \omega_x + \omega_y + \frac{g}{V} \gamma \\ \frac{d\omega_x}{dt} = \bar{M}_x^\beta \beta + \bar{M}_x^{\omega_x} \omega_x + \bar{M}_x^{\omega_y} \omega_y + \bar{M}_x^{\delta_3} \delta_3 \\ \frac{d\omega_y}{dt} = \bar{M}_y^\beta \beta + \bar{M}_y^{\omega_x} \omega_x + \bar{M}_y^{\omega_y} \omega_y + \bar{M}_y^{\delta_H} \delta_H \\ \frac{d\gamma}{dt} = \omega_x \\ \frac{d\Psi}{dt} = \omega_y \end{cases} \quad (1.1.2)$$

Данной системе уравнений соответствует структурная схема на рисунке

1:

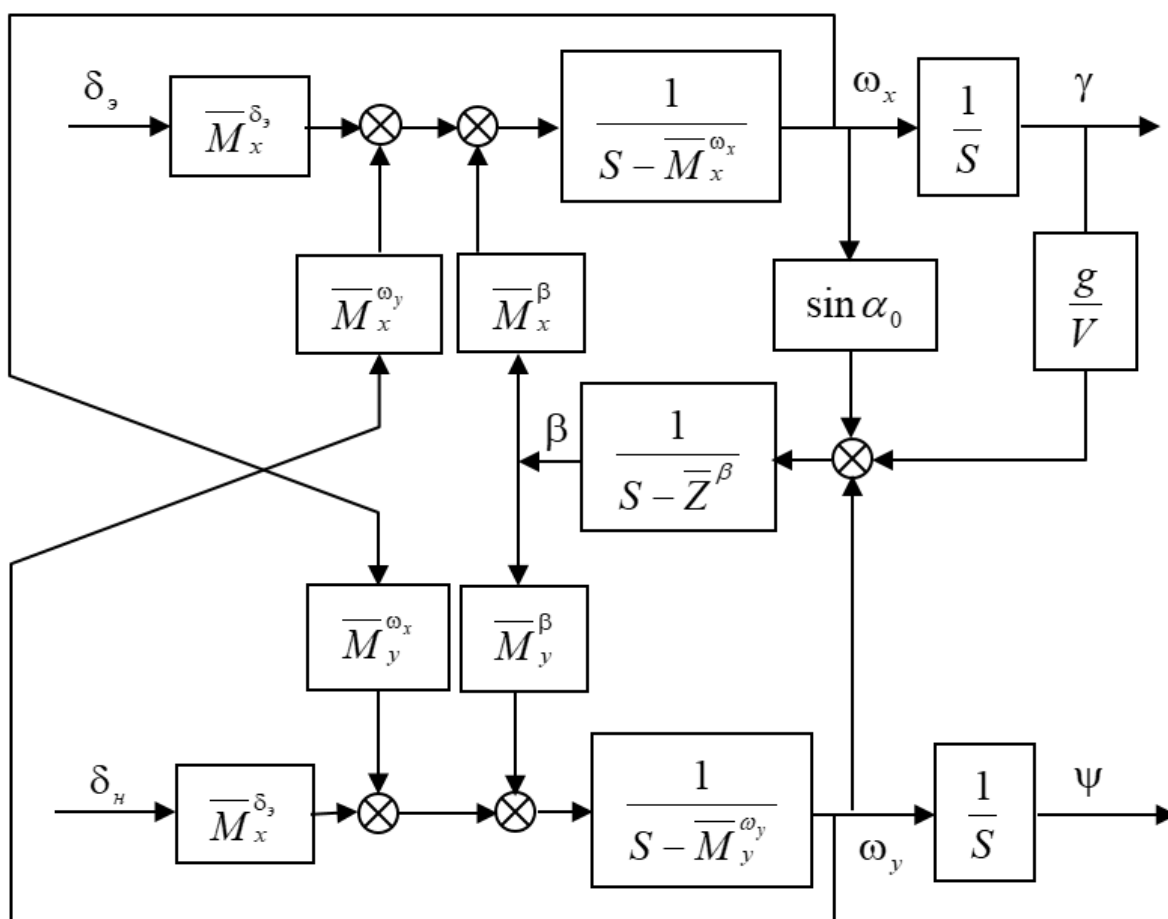


Рисунок 1. Структурная схема бокового движения самолёта

При исследовании неуправляемого бокового движения самолёта принимается $\delta_{\text{э}} = \delta_{\text{н}} = 0$, а последнее уравнение системы (1.1.2) обычно отбрасывается, так как силы и моменты, действующие на самолёт, не зависят от угла рыскания (в первые четыре уравнения ψ не входит). Получаемая при этом из (1.1.2) система имеет характеристический многочлен четвёртого порядка: $\lambda^4 + A_3\lambda^3 + A_2\lambda^2 + A_1\lambda + A_0$ коэффициенты которого выражаются через коэффициенты системы (2) следующим образом:

$$\begin{cases} A_3 = -\bar{Z}^\beta - \bar{M}_x^{\omega_x} - \bar{M}_y^{\omega_y} \\ A_2 = \bar{Z}^\beta \bar{M}_y^{\omega_y} - \bar{M}_y^\beta + \bar{M}_x^{\omega_x} (\bar{Z}^\beta + \bar{M}_y^{\omega_y}) - \bar{M}_x \sin \alpha_0 - \bar{M}_x^{\omega_y} \bar{M}_y^{\omega_x} \\ A_1 = \bar{M}_x^{\omega_x} (\bar{M}_y^\beta - \bar{Z}^\beta \bar{M}_y^{\omega_y}) - \bar{M}_x^\beta \left(\frac{g}{V} + \bar{M}_y^{\omega_x} - \bar{M}_y^{\omega_y} \sin \alpha_0 \right) + \bar{M}_x^{\omega_y} (\bar{Z}^\beta \bar{M}_y^{\omega_x} - \bar{M}_y^\beta \sin \alpha_0) \\ A_0 = \frac{g}{V} (\bar{M}_y^{\omega_y} \bar{M}_x^\beta - \bar{M}_y^\beta \bar{M}_x^{\omega_y}) \end{cases} \quad (1.1.3)$$

Характеристический многочлен имеет два действительных и два комплексно-сопряжённых корня. Один действительный корень всегда большой по модулю, он называется креновым и обозначается $\lambda_{\text{кр}}$, другой действительный корень очень мал по модулю, называется спиральным и обозначается $\lambda_{\text{сп}}$. Модуль комплексных корней находится между модулями кренового и спирального корней. Квадратный трёхчлен, образованный комплексными корнями, обычно обозначается как: $\lambda^2 + 2\xi_6\omega_6\lambda + \omega_6^2$ и, таким образом, весь характеристический многочлен можно представить в виде: $(\lambda - \lambda_{\text{кр}})(\lambda^2 + 2\xi_6\omega_6\lambda + \omega_6^2)(\lambda - \lambda_{\text{сп}})$

В боковом движении может быть два вида неустойчивости: спиральная, когда в правой полуплоскости находится корень $\lambda_{\text{сп}}$, и колебательная, когда там находятся комплексные корни.

1.2. Упрощённые модели бокового движения

Так как спиральный корень по модулю очень мал, то при исследовании креновой и колебательных составляющих без большой погрешности можно принять его равным нулю. Так как физически спиральное движение возникает из-за влияния крена на скольжение, то "обнуление" спирального корня произойдёт при пренебрежении слагаемым $\frac{g}{V}\gamma$ в системе (1.1.2). Часто можно пренебречь спиральными моментами $\bar{M}_x^{\omega_y}$, $\bar{M}_y^{\omega_x}$; а также слагаемыми $\sin \alpha_0 \omega_x$.

В итоге система (1.1.2) принимает вид:

$$\begin{cases} \frac{d\beta}{dt} = \bar{Z}^{\beta} \beta + \omega_y \\ \frac{d\omega_x}{dt} = \bar{M}_x^{\beta} \beta + \bar{M}_x^{\omega_x} \omega_x + \bar{M}_x^{\delta_{\text{э}}} \delta_{\text{э}} \\ \frac{d\omega_y}{dt} = \bar{M}_y^{\beta} \beta + \bar{M}_y^{\omega_y} \omega_y + \bar{M}_y^{\delta_{\text{н}}} \delta_{\text{н}} \\ \frac{d\gamma}{dt} = \omega_x \\ \frac{d\Psi}{dt} = \omega_y \end{cases} \quad (1.2.1)$$

Как видно, движение рыскания (1-ое, 3-е и 5-ое уравнения) стало независимым от движения крена (2-ое и 4-ое уравнения), но не наоборот.

Система (1.2.1) распадается на две системы:

$$\begin{cases} \frac{d\beta}{dt} = \bar{Z}^{\beta} \beta + \omega_y \\ \frac{d\omega_y}{dt} = \bar{M}_y^{\beta} \beta + \bar{M}_y^{\omega_y} \omega_y + \bar{M}_y^{\delta_{\text{н}}} \delta_{\text{н}} \\ \frac{d\Psi}{dt} = \omega_y \end{cases} \quad (1.2.2)$$

$$\begin{cases} \frac{d\omega_x}{dt} = \bar{M}_x^{\beta} \beta + \bar{M}_x^{\omega_x} \omega_x + \bar{M}_x^{\delta_{\text{э}}} \delta_{\text{э}} \\ \frac{d\gamma}{dt} = \omega_x \end{cases} \quad (1.2.3)$$

Вместо β в правую часть первого уравнения системы (1.2.3) следует подставлять решение системы (1.2.2).

Системе уравнений (1.2.1) соответствует структурная схема, показанная на рисунке 2:

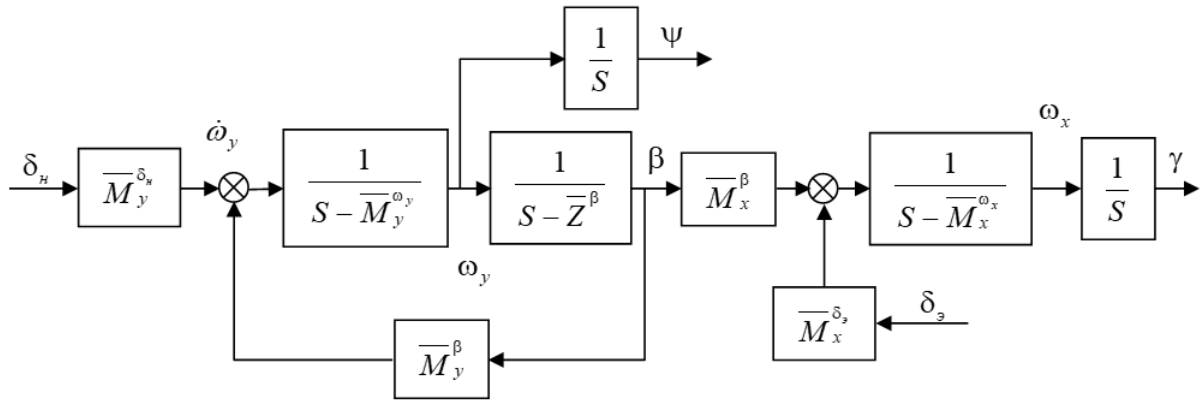


Рисунок 2. Структурная схема по системе (1.2.2)

Как видно из этой схемы, движение крена, возникающее при отклонении элеронов, можно исследовать независимо от движения рыскания, т.е. используя систему (1.2.3) при $\beta = 0$:

$$\begin{cases} \frac{d\omega_x}{dt} = \bar{M}_x^{\omega_x} \omega_x + \bar{M}_x^{\delta_3} \delta_3 \\ \frac{d\gamma}{dt} = \omega_x \end{cases} \quad (1.2.4)$$

Уравнения (1.2.4) называются уравнениями изолированного движения крена, а (1.2.2) – изолированного движения рыскания.

Характеристический многочлен системы (4) имеет вид:

$$(\lambda - \bar{M}_x^{\omega_x})[\lambda^2 + (-\bar{Z}^\beta - \bar{M}_y^{\omega_y})\lambda + (-\bar{M}_y^{\omega_y})] \quad (1.2.5)$$

Трёхчлен в квадратных скобках обозначается, как $\lambda^2 + 2\xi_\beta \omega_\beta \lambda + \omega_\beta^2$, так как он является характеристическим многочленом двух первых уравнений системы (5), описывающих изолированные колебания самолёта по ω_y и β .

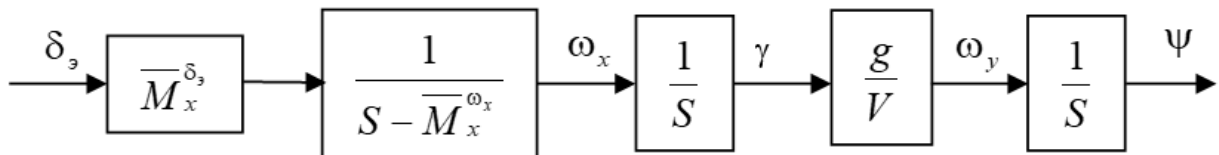


Рисунок 3. Структурная схема по системе (1.2.6)

При исследовании координированного ($\beta = 0$) или близкого к координированному (β мал) развороту применяется система уравнений (1.2.6) которой соответствует структурная схема на рисунке 3.

$$\begin{cases} \omega_y + \frac{g}{V} \gamma = 0 \\ \frac{d\omega_x}{dt} = \bar{M}_x^{\omega_x} \omega_x + \bar{M}_x^{\delta_3} \delta_3 \\ \frac{d\gamma}{dt} = \omega_x \\ \frac{d\Psi}{dt} = \omega_y \end{cases} \quad (1.2.6)$$

2. Система стабилизации угла крена

2.1. Теоретические сведения

Функциональная схема статической системы стабилизации угла крена:

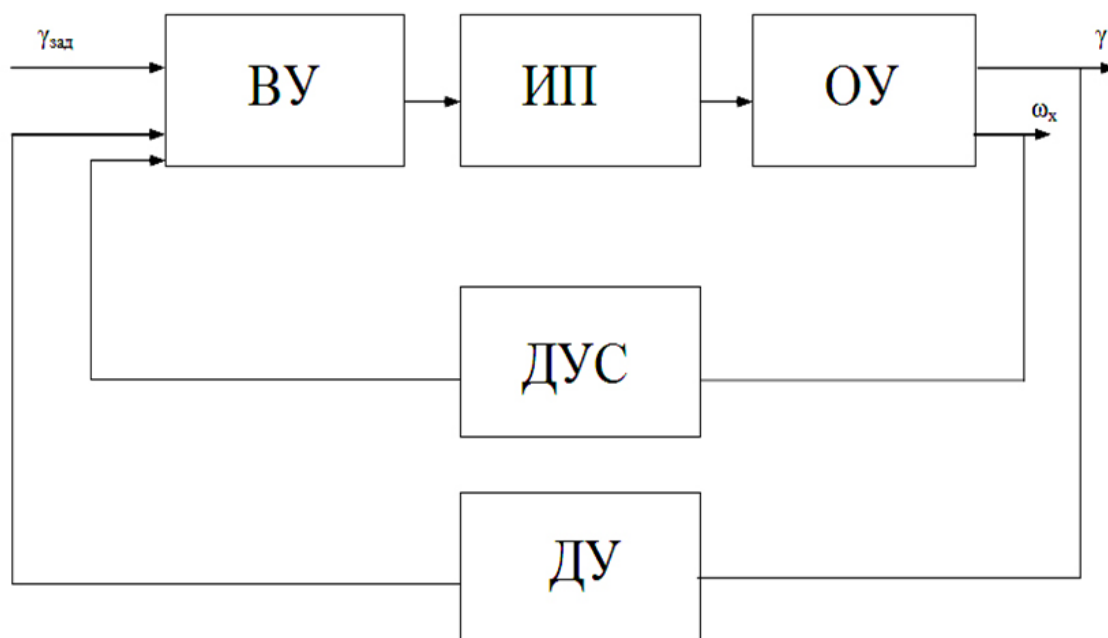


Рисунок 4. Функциональная схема статической системы стабилизации угла крена

где:

ВУ – вычислительное устройство,

ИП – исполнительный привод,

ОУ – объект управления,

ДУ – датчик угла,

ДУС – датчик угловых скоростей.

Закон управления статической системы стабилизации угла крена имеет вид:

$$\delta_{\alpha} = K_{\gamma} (\gamma - \gamma_{\text{зад}}) + K_{\omega_x} \omega_x \quad (2.1.1)$$

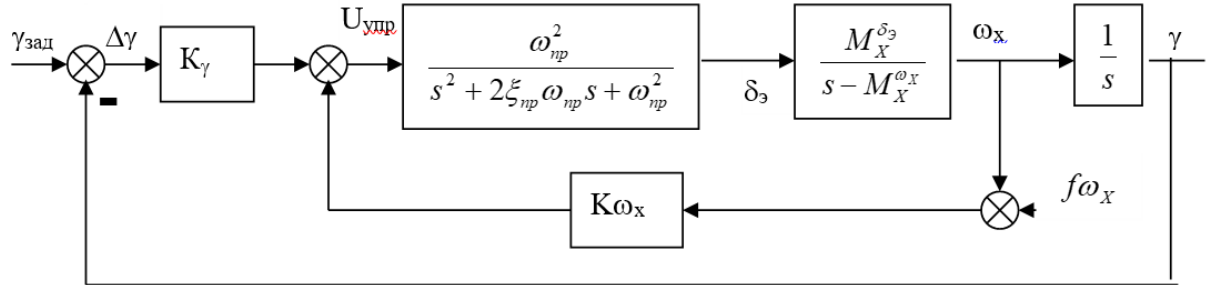


Рисунок 5. Система стабилизации угла крена

$$W_{\text{пр}}(s) = \frac{\omega_{\text{пр}}^2}{s^2 + 2\xi_{\text{пр}} \omega_{\text{пр}} s + \omega_{\text{пр}}^2} = \frac{400}{s^2 + 28,3s + 400} \quad (2.1.2)$$

Для приближённого анализа процессов управления креном и предварительного выбора передаточных чисел САУ можно воспользоваться уравнением изолированного движения крена (1.2.4).

Уравнение (1.2.4) совместно с уравнением САУ (1.2.6) образуют систему уравнений замкнутого контура автоматической стабилизации угла крена. Структурная схема контура представлена на рисунке 4. Передаточная функция самолёта из уравнений (1.2.4) имеет вид:

$$W_{\gamma}^{\delta_{\alpha}}(s) = \frac{\gamma(s)}{\delta_{\alpha}(s)} = \frac{\bar{M}_X^{\delta_{\alpha}}}{s(s - \bar{M}_X^{\omega_x})} \quad (2.1.3)$$

Передаточная функция замкнутого контура в режиме управления запишется следующим образом:

$$W_{\gamma}^{\gamma_{\text{зад}}}(s) = - \frac{-K_{\gamma} \bar{M}_X^{\delta_{\alpha}}}{s^2 + (-K_{\omega_x} \bar{M}_X^{\delta_{\alpha}} - \bar{M}_X^{\omega_x})s - K_{\gamma} \bar{M}_X^{\delta_{\alpha}}} \quad (2.1.4)$$

2.2. Синтез системы управления углом крена

Передаточная функция (2.1.4) представляет собой передаточную функцию колебательного звена. Поэтому выбор передаточных чисел K_γ и K_{ω_x} можно достаточно просто произвести с помощью метода стандартных коэффициентов.

Представим передаточную функцию (2.1.4) в виде:

$$W_\gamma^{\gamma_{\text{зад}}}(s) = \frac{\omega_0^2}{s^2 + 2\xi\omega_0 s + \omega_0^2} \quad (2.2.1)$$

где:

$$\begin{cases} 2\xi\omega_0 = -K_{\omega_x}\bar{M}_X^{\delta_\partial} - \bar{M}_X^{\omega_x} \\ \omega_0^2 = -K_\gamma\bar{M}_X^{\delta_\partial} \end{cases} \quad (2.2.2)$$

Качество переходного процесса полностью определяется степенью демпфирования ξ и собственной частотой ω_0 .

Характеристическое уравнение:

$$\lambda^2 + 2\xi\omega_0\lambda + \omega_0^2 = 0 \quad (2.2.3)$$

Решая его, получим

$$\lambda_{1,2} = \frac{-2\xi\omega_0 \pm \sqrt{4\omega_0^2(\xi^2 - 1)}}{2} = -\xi\omega_0 \pm \omega_0\sqrt{(\xi^2 - 1)} \quad (2.2.4)$$

При этом

$\delta = \xi\omega_0$ – коэффициент затухания

$\omega_c = \omega_0\sqrt{(1 - \xi^2)}$ – собственная частота колебаний

$T = \frac{2\pi}{\omega_c} = \frac{2\pi}{\omega_0\sqrt{(1 - \xi^2)}}$ – период колебаний

$\tau = \frac{1}{\delta} = \frac{1}{\xi\omega_0}$ – время релаксации

Время переходного процесса будет примерно равно

$$t_p \approx 4\tau \quad (2.2.5)$$

При этом $t_p = 4 \frac{1}{\xi \omega_0}$, $T = \frac{2\pi}{\omega_0 \sqrt{(1-\xi^2)}}$

При этом, чтобы перерегулирование не было больше 5%, должно соблюдаться условие $0 < 2t_p \leq T$. При этом необходимо минимизировать время переходного процесса. Для этого нужно максимизировать ω_0 и ξ .

Решим неравенство

$$0 < 2t_p \leq T \quad (2.2.6)$$

$$0 < \frac{8}{\xi \omega_0} \leq \frac{2\pi}{\omega_0 \sqrt{(1-\xi^2)}} \quad (2.2.7)$$

Преобразуем крест-накрест и поделим левую и правую часть на $2\omega_0$:

$$0 < 4\sqrt{(1-\xi^2)} \leq \pi\xi \quad (2.2.8)$$

Возведём левую и правую часть в квадрат:

$$0 < 16(1-\xi^2) \leq \pi^2 \xi^2 \quad (2.2.9)$$

$$0 < \frac{16}{\pi^2 + 16} \leq \xi^2 \quad (2.2.10)$$

Отсюда

$$\xi \in \left[\sqrt{\frac{16}{\pi^2 + 16}}; +\infty \right)$$

Возьмём $\xi = \sqrt{\frac{16}{4\pi^2 + 16}} = 0.7864$

Как видно, перерегулирование не зависит от параметра ω_0 .

$$\omega_0 = \frac{4}{\xi t_p} \quad (2.2.11)$$

Если задать $t_p = 1$ с, то $\omega_0 = 5.0862$

Подставляя полученные значения ω_0 и ξ , находим передаточные числа K_γ и K_{ω_x} , соответствующие оптимальной переходной функции:

$$K_\gamma = -\frac{\omega_0^2}{\bar{M}_X \delta_3} = 3.6957 \quad (2.2.12)$$

$$K_{\omega_x} = -\frac{(\bar{M}_X^{\omega_x} + 2\xi\omega_0)}{\bar{M}_X \delta_3} = 1 \quad (2.2.13)$$

Исходная передаточная функция системы стабилизации угла крена без учёта привода примет вид:

$$W_\gamma^{\gamma_{\text{зад}}}(s) = \frac{-25.87}{s^2 + 8s + 25.87} \quad (2.2.14)$$

Исходная передаточная функция системы стабилизации угла крена с учётом привода примет вид:

$$W_\gamma^{\gamma_{\text{зад}}}(s) = -\frac{10350}{s^4 + 29.28s^3 + 428.3s^2 + 3200s + 10350} \quad (2.2.15)$$

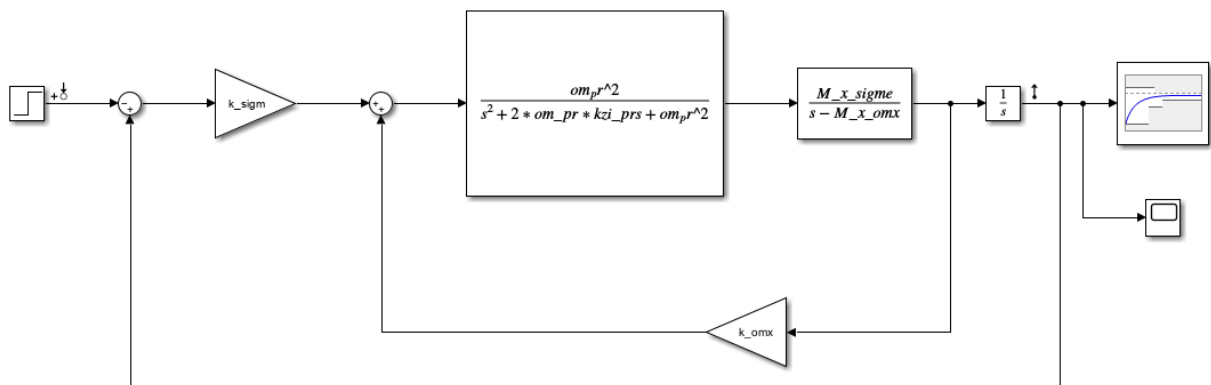


Рисунок 6. Схема моделирования САУ крена в среде Simulink с учетом динамики привода

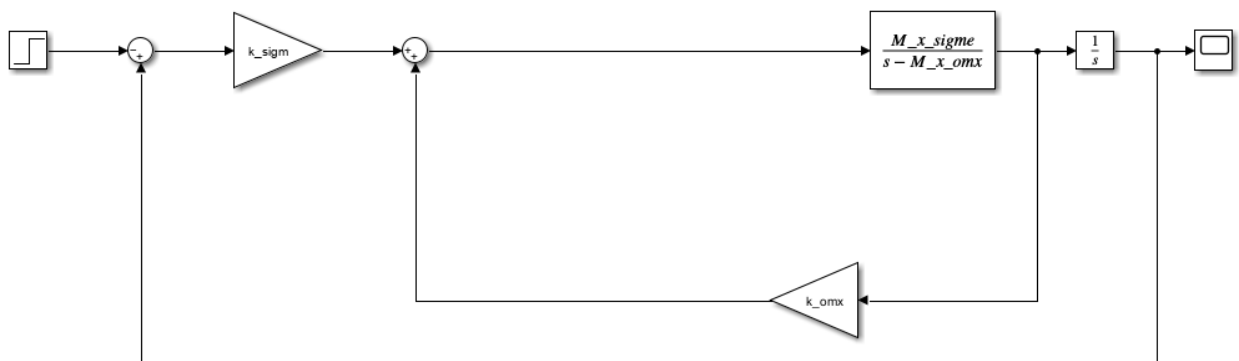


Рисунок 7. Схема моделирования САУ крена в среде Simulink без учета динамики привода

График переходных функций в системе с учётом привода и без учёта привода представлен на рисунке 8.

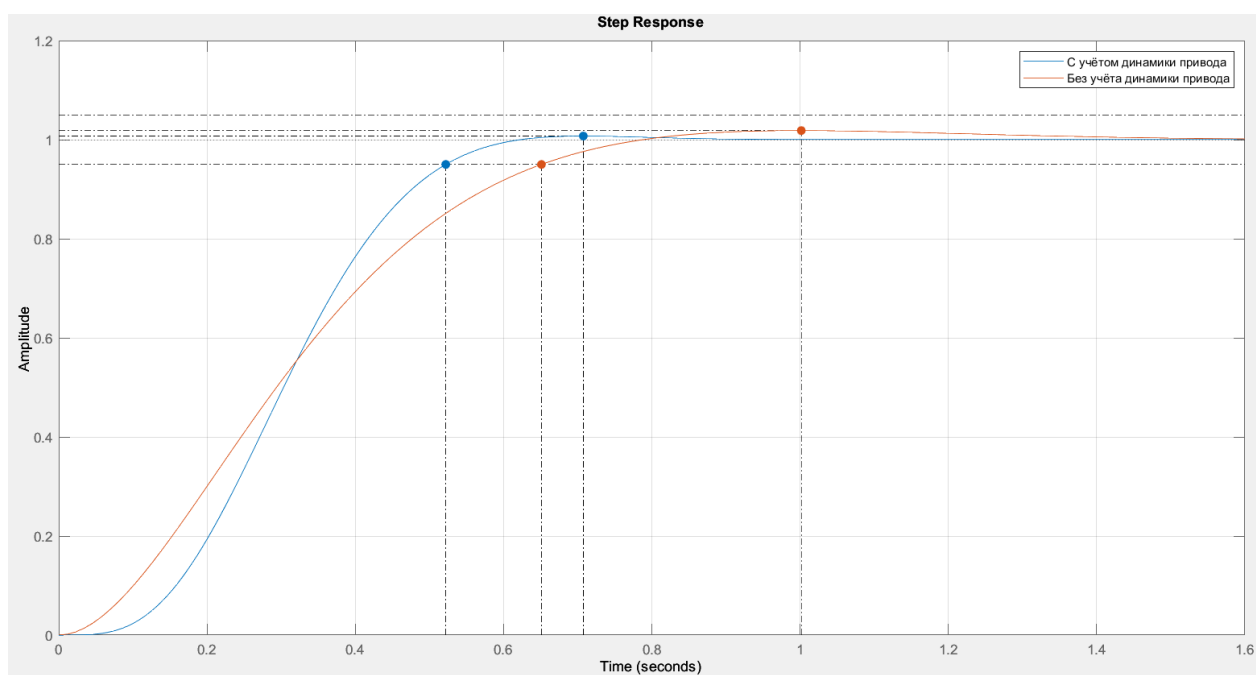


Рисунок 8. График переходной функции для системы стабилизации крена

С учетом динамики привода: $t_{п.п.} = 0.522$; $h_{max} = 1.01$; $\sigma = 0.727\%$

Без учета динамики привода: $t_{п.п.} = 0.651$; $h_{max} = 1.02$; $\sigma = 1.83\%$

2.3. Подбор коэффициентов в NCD-блоке для САУ углом крена

Теперь попробуем воспользоваться NCD-блоком в Simulink для подбора коэффициентов, чтобы уменьшить длительность переходных процессов. Схема с NCD-блоком представлена на рисунке 9.

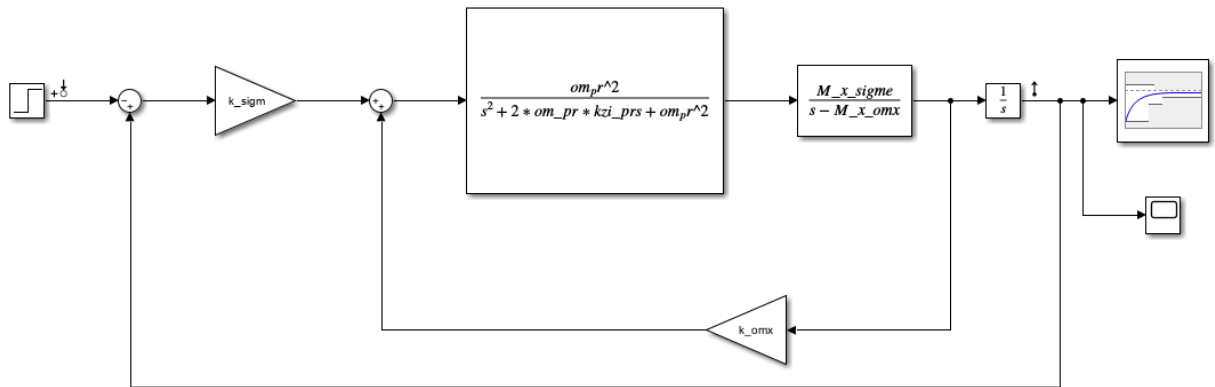


Рисунок 9. Схема моделирования САУ крена с учётом динамики привода с NCD-блоком

В результате многократных испытаний получаем переходный процесс с минимальным временем переходного процесса.

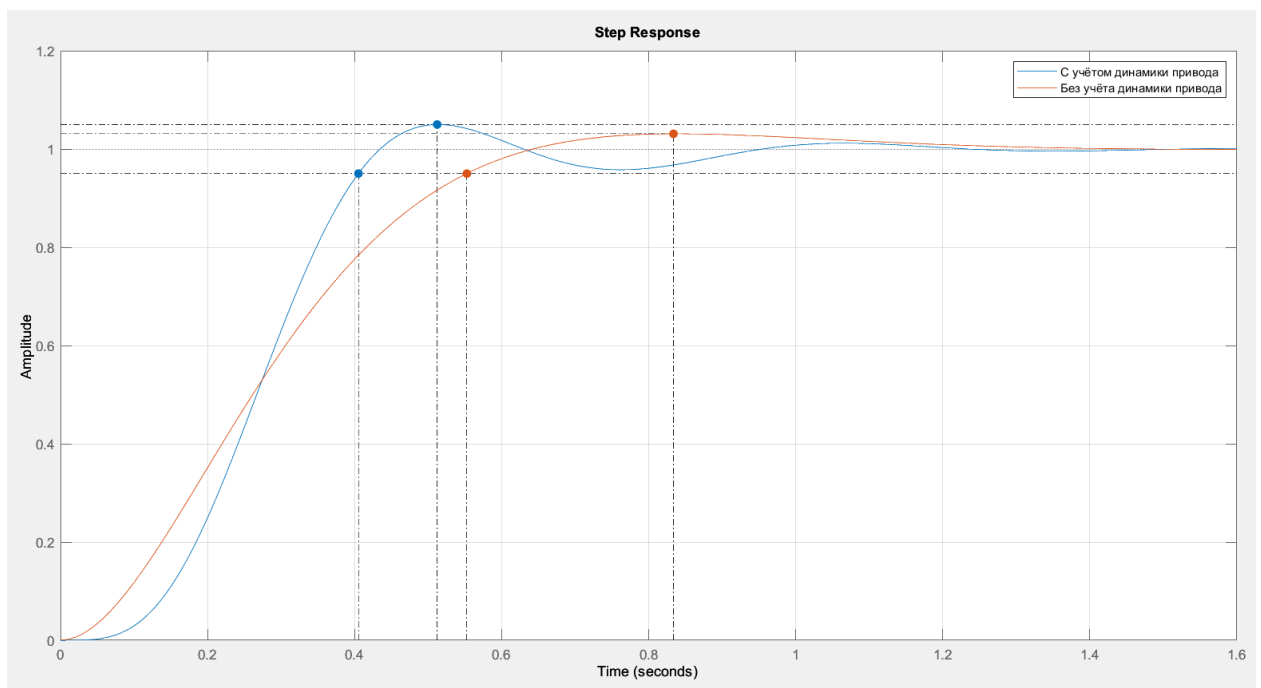


Рисунок 10. График переходного процесса по крену после моделирования в NCD-блоке

С учетом динамики привода: $t_{п.п.} = 0.4$; $h_{max} = 1.03$; $\sigma = 3.33\%$

Без учета динамики привода: $t_{п.п.} = 0.547$; $h_{max} = 1.02$; $\sigma = 1.59\%$

Данные результаты получились при моделировании с коэффициентами равными

$$K_\gamma = 5.4231 \quad (2.3.1)$$

$$K_{\omega_x} = 1.2597 \quad (2.3.2)$$

Исходная передаточная функция системы стабилизации угла крена без учёта привода примет вид:

$$W_\gamma^{\gamma_{\text{зад}}}(s) = -\frac{37.96}{s^2 + 9.818s + 37.96} \quad (2.3.3)$$

Исходная передаточная функция системы стабилизации угла крена с учётом привода примет вид:

$$W_\gamma^{\gamma_{\text{зад}}}(s) = -\frac{15180}{s^4 + 29.28s^3 + 428.3s^2 + 3927s + 15180} \quad (2.3.4)$$

Как видно, моделирование в NCD-блоке позволило достичь минимального времени переходного процесса, подобрав соответствующие коэффициенты, но при этом видно, что качество переходного процесса в системе с учётом динамики привода заметно уменьшилось.

2.4. Построение области устойчивости для САУ углом крена

Построим область устойчивости в зависимости от параметров K_Y и K_{ω_x} . Для этого воспользуемся критерием Рауса. Для системы четвёртого порядка условия устойчивости по этому критерию выглядят следующим образом:

$$\begin{cases} A_4 > 0 \\ A_3 > 0 \\ A_2 > 0 \\ A_1 > 0 \\ A_0 > 0 \\ A_1 A_2 A_3 - A_1^2 A_4 - A_0 A_3^2 > 0 \end{cases} \quad (2.4.1)$$

Где A_0, A_1, A_2, A_3, A_4 – коэффициенты характеристического многочлена полной системы с учётом привода.

$$\begin{cases} A_4 = 1 \\ A_3 = 29.28 \\ A_2 = 428.3 \\ A_1 = 2800K_{\omega_x} + 400 \\ A_0 = 2800.0K_Y \end{cases} \quad (2.4.2)$$

Область устойчивости – это область, которая соответствует выполнению критерия Рауса для системы 4 порядка в данном случае, и на рисунке 11 она обведена оранжевым цветом.

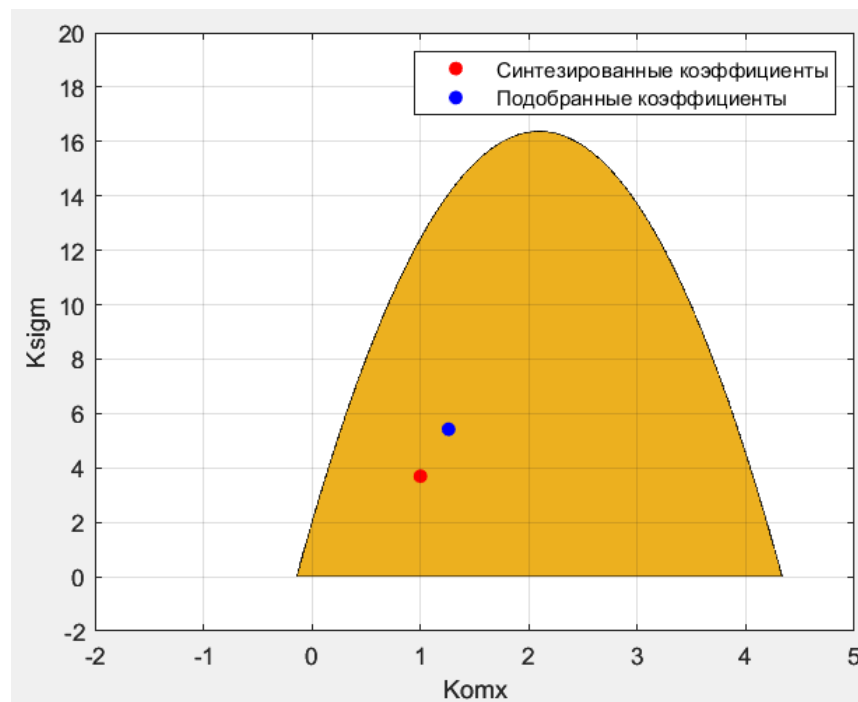


Рисунок 11. Область устойчивости для системы управления креном

2.5. Построение области устойчивости при использовании достаточных условий устойчивости для САУ углом крена и сравнение такого метода с методом построения по критерию Рауса

Построим область устойчивости в зависимости от параметров K_Y и K_{ω_x} , используя достаточные условия устойчивости. Для системы четвёртого порядка условия устойчивости по этому критерию выглядят следующим образом:

$$\begin{cases} \lambda_1 = A_0^{-1}A_1A_2A_3^{-1} \geq 2.15 \\ \lambda_2 = A_1^{-1}A_2A_3A_4^{-1} \geq 2.15 \end{cases} \quad (2.5.1)$$

Где A_0, A_1, A_2, A_3, A_4 – коэффициенты характеристического многочлена полной системы с учётом привода (система (2.4.2)).

Область устойчивости, удовлетворяющая системе (2.5.1) представлена на рисунке 12.

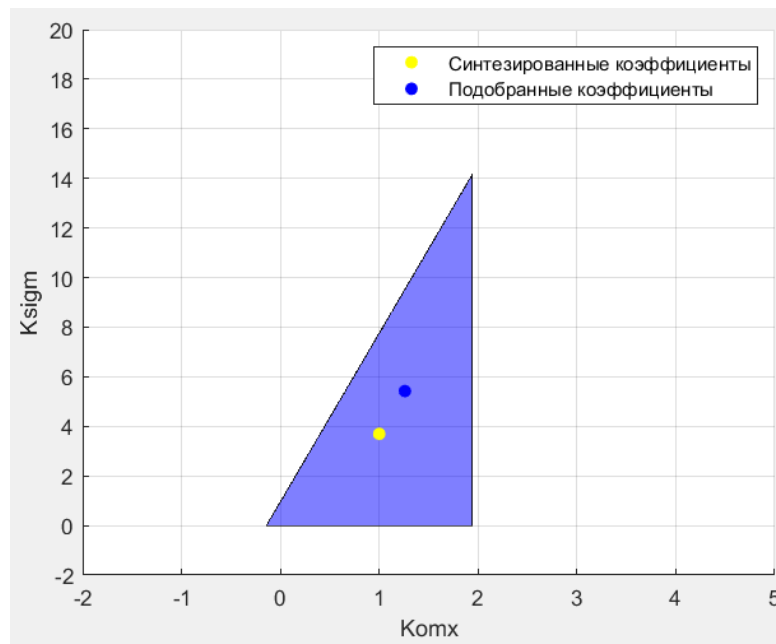


Рисунок 12. Область устойчивости для САУ крена по достаточным условиям устойчивости

Теперь поверх этой области построим область устойчивости, определяемую условиями по критерию Рауса (глава 2.4). Данная область показана на рисунке 13.

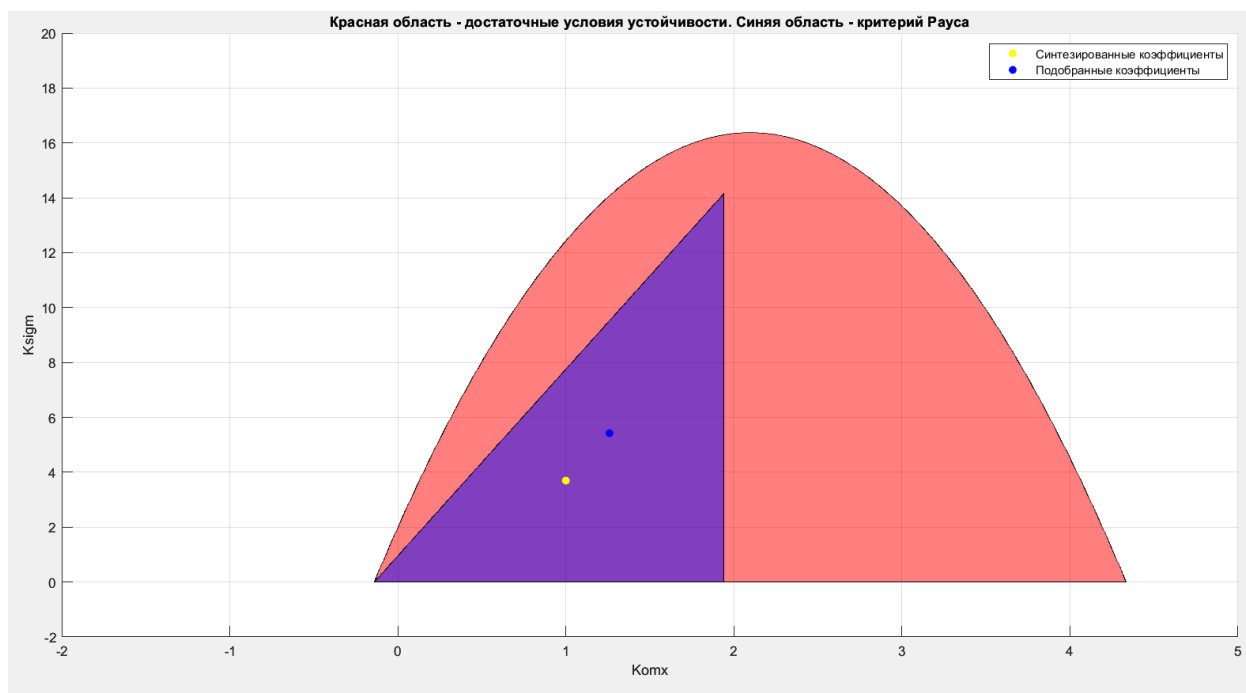


Рисунок 13. Области устойчивости по двум критериям для САУ крена

Видно, что область устойчивости, построенная по критерию достаточного условия устойчивости (синяя), полностью лежит внутри области, построенной по критерию Рауса (красная).

Площадь синей области примерно в 3 раза меньше красной области, кроме того неравенств, по которым строится красная область 6 для данного случая, а неравенств, по которым строится синяя область 2 для данного случая. Это означает, что есть возможность уменьшить скорость построения области по достаточным условиям устойчивости примерно в 9 раз.

Также видно, что большая часть точек синей области лежит условно далеко от границы красной области, а значит, коэффициенты, выбранные внутри синей области почти всегда будут давать качественный переходный процесс.

3. Система стабилизации угла рыскания в режиме плоского разворота

3.1. Теоретические сведения

Плоский разворот – управление курсом воздействием на руль направления. При таком способе руль направления создаёт скольжение, которое в свою очередь создаёт поперечную, искривляющую траекторию силу $\bar{Z}^{\beta} \beta$. Для осуществления плоского разворота необходимо отклонять элероны для стабилизации угла крена (для ликвидации моментов $\bar{M}_x^{\beta} \beta$ и $\bar{M}_x^{\omega_y} \omega$). Т.е. в плоском развороте руль направления используется для создания скольжения, а элероны – для ликвидации крена.

Закон управления идеальной статической системы стабилизации угла рыскания имеет вид:

$$\delta_n = K_{\psi}(\Psi - \Psi_{\text{зад}}) + K_{\omega_y} \omega_y \quad (3.1.1)$$

Для приближённого анализа движения рыскания и предварительного выбора передаточных чисел можно воспользоваться уравнениями изолированного движения рыскания (1.2.2). Добавив к уравнениям (1.2.2) уравнение САУ (3.1.1), получим систему уравнений замкнутого контура. Функциональная схема системы стабилизации угла рыскания в режиме плоского разворота:

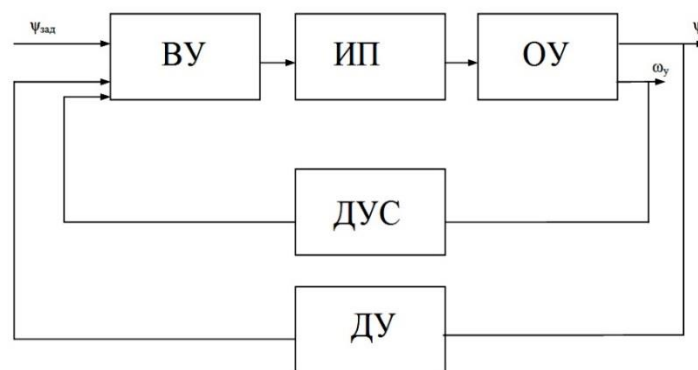


Рисунок 14. Функциональная схема системы стабилизации угла рыскания в режиме плоского разворота

Структурная схема контура показана на рисунке 15.

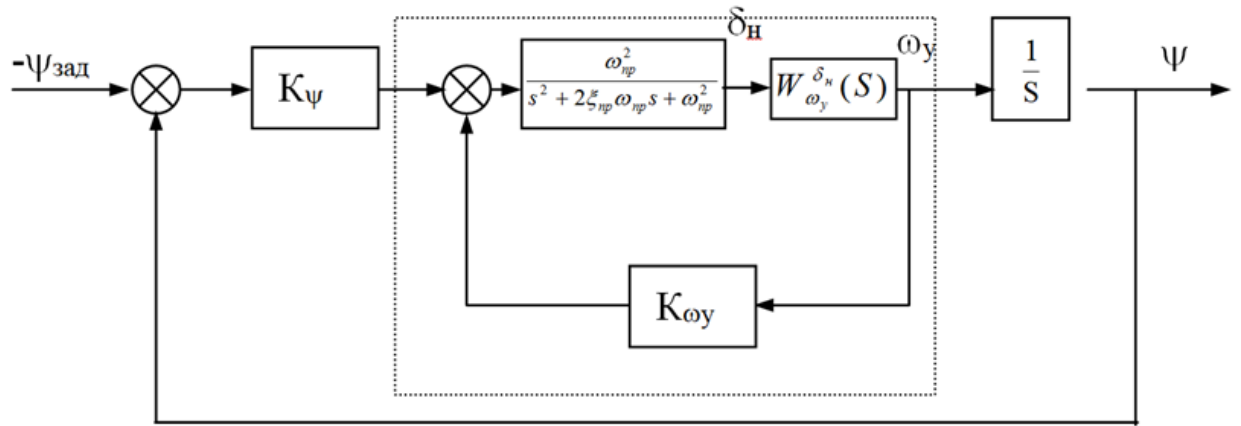


Рисунок 15. Структурная схема управления углом рыскания в режиме плоского разворота

Передаточная функция самолёта по угловой скорости рыскания, полученная из уравнений (5), имеет вид:

$$W_{\omega_y}^{\delta_n}(s) = \frac{\Psi(s)}{\delta_n(s)} = \frac{\bar{M}_y^{\delta_n} s - \bar{M}_y^{\delta_n} \bar{Z}^{\beta}}{s^2 + (-\bar{M}_y^{\omega_y} - \bar{Z}^{\beta})s + (\bar{Z}^{\beta} \bar{M}_y^{\omega_y} - \bar{M}_y^{\beta})} \quad (3.1.2)$$

Передаточная функция для системы на рисунке 11 определяется формулой:

$$W_{\psi}^{-\psi_{\text{зад}}}(s) = \frac{K_{\psi} \frac{W_{\omega_y}^{\delta_n} W_{\text{пр}}}{1 - W_{\omega_y}^{\delta_n} W_{\text{пр}} K_{\omega_y}} \frac{1}{s}}{1 - K_{\psi} \frac{W_{\omega_y}^{\delta_n} W_{\text{пр}}}{1 - W_{\omega_y}^{\delta_n} W_{\text{пр}} K_{\omega_y}} \frac{1}{s}} \quad (3.1.3)$$

$$W_{\psi}^{-\psi_{\text{зад}}}(s) = \frac{-1000K_{\psi}s - 200K_{\psi}}{s^5 + 28.68s^4 + 414.35s^3 + (1000K_{\omega_y} + 245.98)s^2 + (1000K_{\psi} + 200K_{\omega_y} + 1216)s + 200K_{\psi}} \quad (3.1.4)$$

3.2. Синтез системы управления углом рыскания в режиме плоского разворота

Коэффициенты характеристического многочлена ПФ 14:

$$\begin{cases} A_5 = 1 \\ A_4 = 28.68 \\ A_3 = 414.35 \\ A_2 = 1000K_{\omega_y} + 245.98 \\ A_1 = 1000K_{\psi} + 200K_{\omega_y} + 1216 \\ A_0 = 200K_{\psi} \end{cases} \quad (3.2.1)$$

Сформулируем задачу математического программирования

$$\min(z) = \min(A_1 A_0^{-1}) \quad (3.2.2)$$

При ограничениях

$$\begin{cases} \lambda_1 = A_0^{-1} A_1 A_2 A_3^{-1} \geq 2.15 \\ \lambda_2 = A_1^{-1} A_2 A_3 A_4^{-1} \geq 2.15 \\ \lambda_3 = A_2^{-1} A_3 A_4 A_5^{-1} \geq 2.15 \\ \delta_1 = A_0^{-1} A_1^2 A_2^{-1} \geq 2 \\ \delta_2 = A_1^{-1} A_2^2 A_3^{-1} \geq 2 \\ \delta_3 = A_2^{-1} A_3^2 A_4^{-1} \geq 2 \\ 0.1 \leq K_{\psi} \leq 300 \\ 0.1 \leq K_{\omega_y} \leq 300 \end{cases} \quad (3.2.3)$$

После определения целевой функции и нелинейных ограничений, применим функцию `fmincon()`. Полученные коэффициенты:

$$K_{\omega_y} = 2.7468 \quad (3.2.4)$$

$$K_{\psi} = 9.0425 \quad (3.2.5)$$

Получили передаточную функцию:

$$W_{\Psi}^{-\Psi_{\text{зад}}} (s) = - \frac{9042.47s + 1808.49}{s^5 + 28.68s^4 + 414.35s^3 + 2992.74s^2 + 10807.82s + 1808.49} \quad (3.2.6)$$

Переходный процесс с заданными коэффициентами показан на рисунке

16.

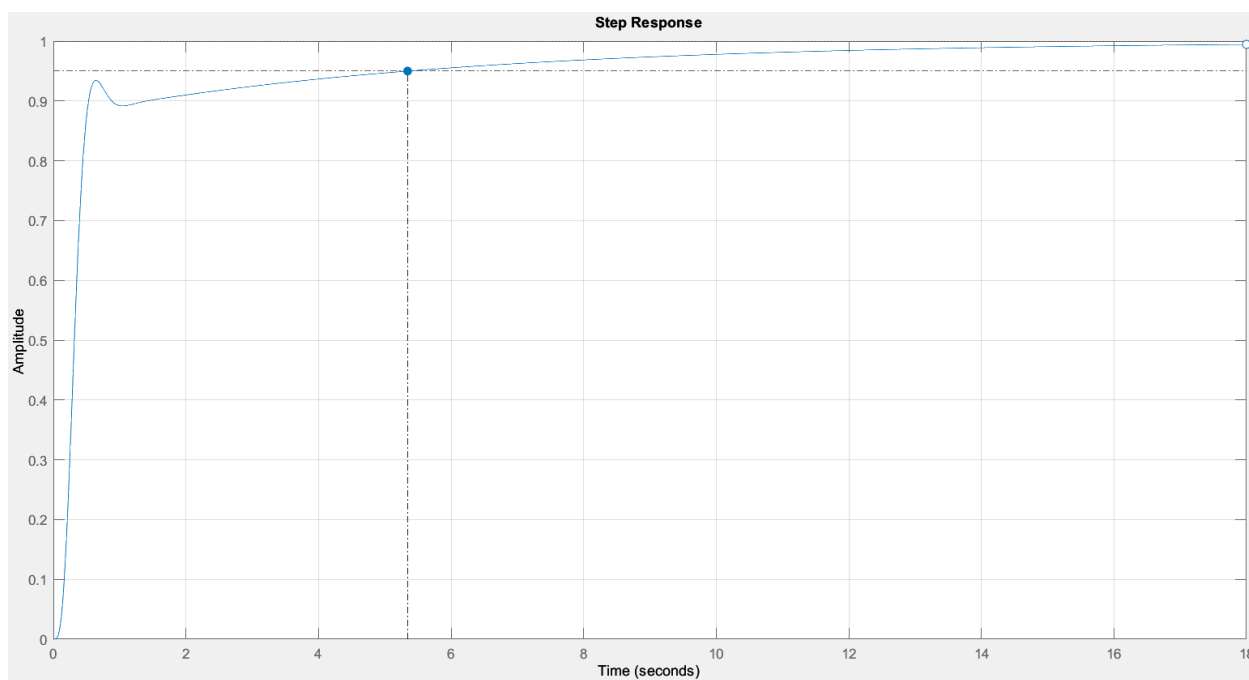


Рисунок 16. Переходный процесс в канале рыскания при плоском развороте после синтеза функцией `fmicon`

$$t_{п.п.} = 5.34; h_{max} = 1; \sigma = 0\%$$

3.3. Подбор коэффициентов в NCD-блоке для САУ углом рыскания в режиме плоского разворота

Осуществим подбор коэффициентов в системе Simulink при помощи NCD-блока для с параметрами моделирования.

$$t_{п.п.} = 3.5; h_{max} = 1.05; \sigma = 5\%$$

Схема моделирования представлена на рисунке 17.

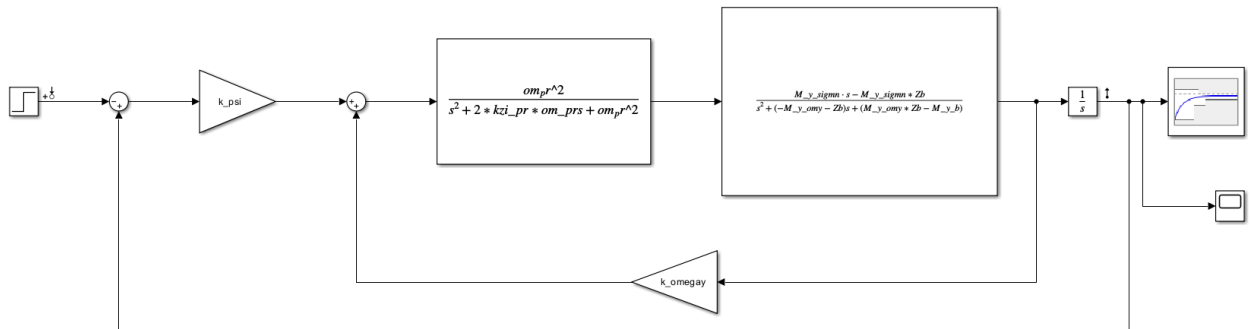


Рисунок 17. Схема моделирования для САУ углом рыскания в режиме плоского разворота

В ходе моделирования получаем коэффициенты

$$K_{\omega_y} = 2.8172 \quad (3.3.1)$$

$$K_{\psi} = 9.0425 \quad (3.3.2)$$

Переходный процесс при таких параметрах показан на рисунке 18.

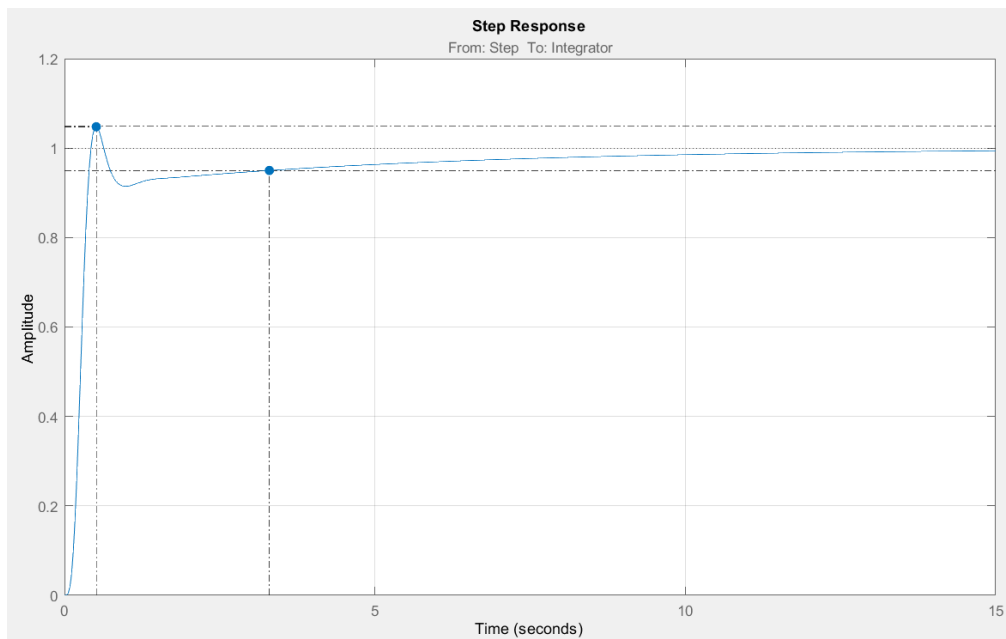


Рисунок 18. Переходный процесс в канале рыскания при плоском развороте после моделирования в NCD-блоке

$$t_{п.п.} = 3.5; h_{max} = 1.05; \sigma = 4.77\%$$

3.4. Построение области устойчивости для САУ углом рыскания в режиме плоского разворота

Построим область устойчивости в зависимости от параметров K_{ω_y} и K_{ψ} . Для этого воспользуемся критерием Рауса. Для системы четвёртого порядка условия устойчивости по этому критерию выглядят следующим образом:

$$\begin{cases} A_5 > 0 \\ A_4 > 0 \\ A_3 > 0 \\ A_2 > 0 \\ A_1 > 0 \\ A_0 > 0 \\ (A_1A_2 - A_0A_3)(A_3A_4 - A_2A_5) - (A_1A_4 - A_0A_5)^2 > 0 \end{cases} \quad (3.4.1)$$

Где $A_0, A_1, A_2, A_3, A_4, A_5$ – коэффициенты характеристического многочлена полной системы с учётом привода (уравнение 3.2.1).

Область устойчивости – это область, которая соответствует выполнению критерия Рауса для системы 5 порядка в данном случае, и на рисунке 19 она обведена оранжевым цветом.

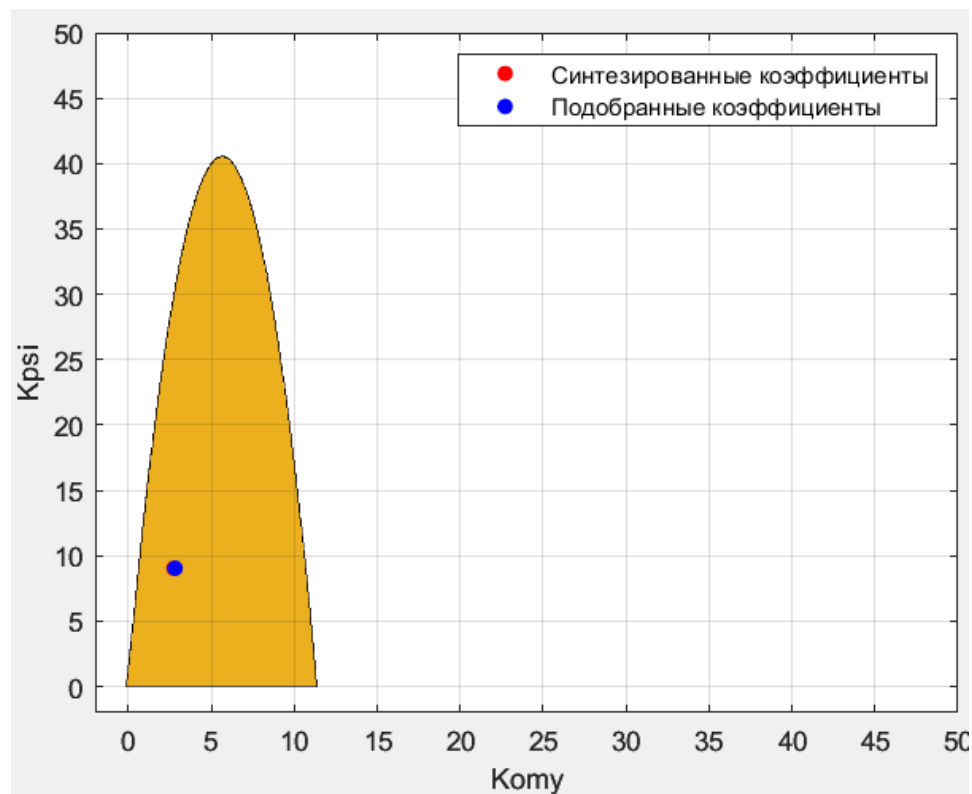


Рисунок 19. Область устойчивости для системы управления рысканием в режиме плоского разворота

3.5. Построение области устойчивости при использовании достаточных условий устойчивости для САУ рыскания при плоском развороте и сравнение такого метода с методом построения по критерию Рауса

Построим область устойчивости в зависимости от параметров K_{ω_y} и K_{ψ} , используя достаточные условия устойчивости. Для системы пятого порядка условия устойчивости по этому критерию выглядят следующим образом:

$$\begin{cases} \lambda_1 = A_0^{-1}A_1A_2A_3^{-1} \geq 2.15 \\ \lambda_2 = A_1^{-1}A_2A_3A_4^{-1} \geq 2.15 \\ \lambda_3 = A_2^{-1}A_3A_4A_5^{-1} \geq 2.15 \end{cases} \quad (3.5.1)$$

Где $A_0, A_1, A_2, A_3, A_4, A_5$ – коэффициенты характеристического многочлена полной системы с учётом привода (система (3.2.1)).

Область устойчивости, удовлетворяющая системе (3.5.1) представлена на рисунке 20.

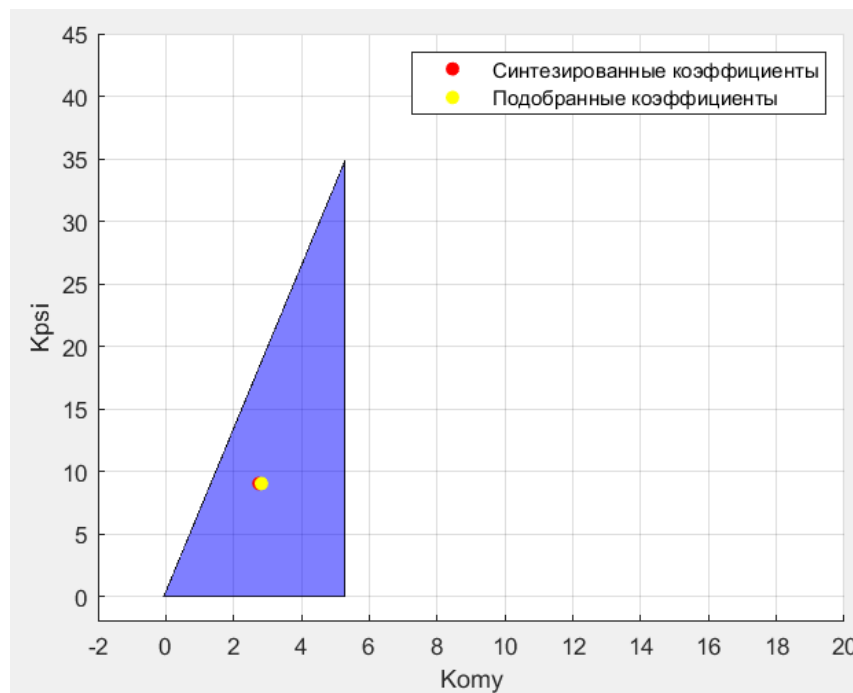


Рисунок 20. Построение области устойчивости по достаточным условиям устойчивости для САУ угла рыскания при плоском развороте

Теперь поверх этой области построим область устойчивости, определяемую условиями по критерию Рауса (глава 3.4). Данная область показана на рисунке 21.

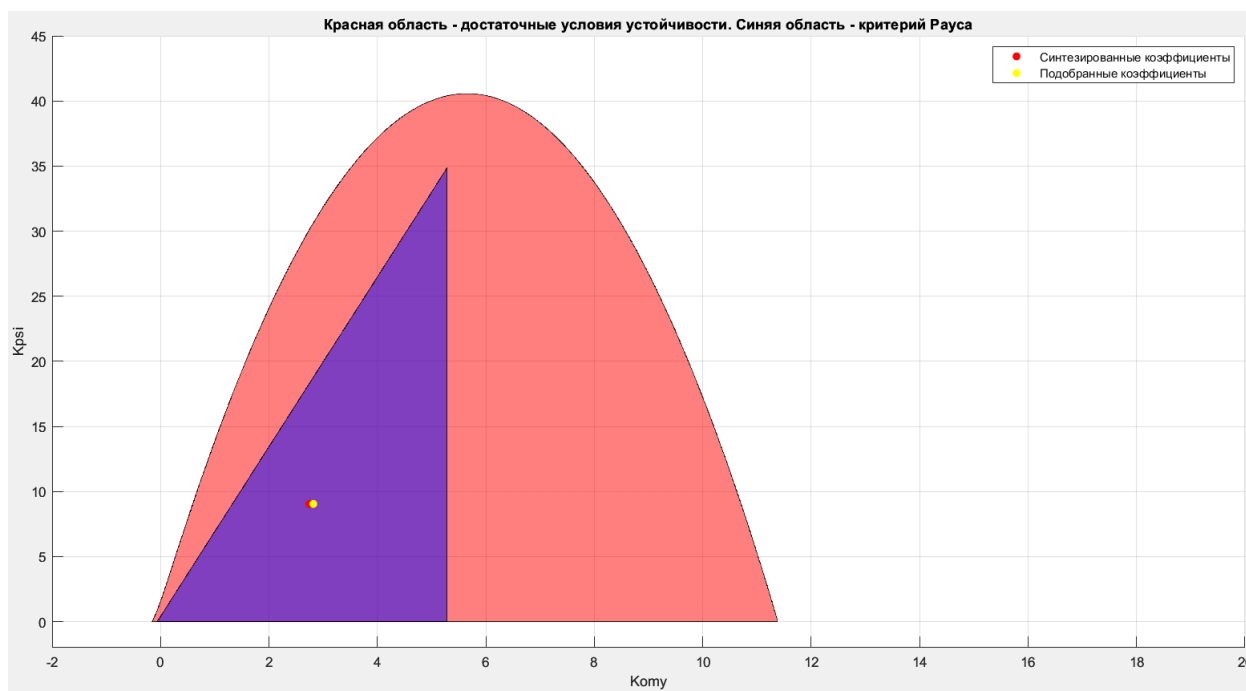


Рисунок 21. Области устойчивости по двум критериям для САУ углом рыскания при плоском развороте

Видно, что область устойчивости, построенная по критерию достаточного условия устойчивости (синяя), полностью лежит внутри области, построенной по критерию Рауса (красная).

Площадь синей области примерно в 3 раза меньше красной области, кроме того неравенств, по которым строится красная область 7 для данного случая, а неравенств, по которым строится синяя область 3 для данного случая. Это означает, что есть возможность уменьшить скорость построения области по достаточным условиям устойчивости примерно в 7 раз.

Также видно, что большая часть точек синей области лежит условно далеко от границы красной области, а значит, коэффициенты, выбранные внутри синей области почти всегда будут давать качественный переходный процесс.

4. Система стабилизации угла рыскания при использовании координированного разворота

4.1. Теоретические сведения

Управление курсом воздействием на элероны называется разворотом с креном. При таком способе разворота в качестве силы, разворачивающей вектор скорости в горизонтальной плоскости, используется проекция подъёмной силы на эту плоскость. Разворот с креном без скольжения называется координированным. Для ликвидации (или уменьшения) скольжения используется руль направления. Таким образом, при развороте с креном руль направления используется для ликвидации скольжения, а элероны – для создания крена.

При развороте с креном сигнал отклонения от заданного курса подаётся не на руль направления, а на элероны. Законы управления каналов элеронов и руля направления идеальной статической САУ принимают вид:

$$\delta_{\varepsilon} = K_{\psi}^{\varepsilon}(\Psi - \Psi_{\text{зад}}) + K_{\gamma}\gamma + K_{\omega_x}\omega_x, \quad \delta_{\text{н}} = K_{\omega_y}\omega_y \quad (4.1.1)$$

При координированном развороте в канал руля направления САУ подаётся сигнал перекрёстной связи по крену. Закон управления (4.1.1) принимает вид:

$$\delta_{\varepsilon} = K_{\psi}^{\varepsilon}(\Psi - \Psi_{\text{зад}}) + K_{\gamma}\gamma + K_{\omega_x}\omega_x \quad (4.1.2)$$

$$\delta_{\text{н}} = K_{\omega_y}\omega_y + K_{\gamma}^{\text{н}}\gamma \quad (4.1.3)$$

Передаточные числа K_{γ} и K_{ω_x} определяются так, как описано во главе 2. Передаточное число $K_{\gamma}^{\text{н}}$ выбирается из условия $\beta = 0$ на всех этапах разворота. Для его выбора используется система уравнений (1.1.2), дополненная законами управления (4.1.2), (4.1.3).

Полученную систему уравнений представляют в области изображения по Лапласу

$$\begin{cases} \beta(s - \bar{Z}^\beta) - \omega_x \sin \alpha_0 - \frac{g}{V} \gamma = 0 \\ \beta(-\bar{M}_x^\beta) + \omega_x(s - \bar{M}_x^{\omega_x} - \bar{M}_x^{\delta_\omega} K_{\omega_x}) - \omega_y \bar{M}_x^{\omega_y} - \gamma K_\gamma \bar{M}_x^{\delta_\gamma} - \psi \bar{M}_x^{\delta_\psi} K_\psi = -\Psi_{\text{зад}} \\ \beta(-\bar{M}_y^\beta) - \omega_x \bar{M}_y^{\omega_x} + \omega_y(s - \bar{M}_y^{\omega_y} - \bar{M}_y^{\delta_\omega} K_{\omega_y}) - \psi K_\psi \bar{M}_y^{\delta_\psi} = 0 \\ -\omega_x + \gamma s = 0 \\ -\omega_y + \psi s = 0 \end{cases} \quad (4.1.4)$$

или $A \cdot X = B \cdot \Psi_{\text{зад}}$, где A и B - матрицы коэффициентов:

$$A = \begin{vmatrix} (s - \bar{Z}^\beta) & -\sin \alpha & -1 & -\frac{g}{V} & 0 \\ -\bar{M}_x^\beta & (s - \bar{M}_x^{\omega_x} - \bar{M}_x^{\delta_\omega} K_{\omega_x}) & -\bar{M}_x^{\omega_y} & -K_\gamma \cdot \bar{M}_x^{\delta_\gamma} & -\bar{M}_x^{\delta_\psi} K_\psi \\ -\bar{M}_y^\beta & -\bar{M}_y^{\omega_x} & (s - \bar{M}_y^{\omega_y} - \bar{M}_y^{\delta_\omega} K_{\omega_y}) & M_y^{\delta_\psi} \cdot k_\gamma & 0 \\ 0 & -1 & 0 & S & 0 \\ 0 & 0 & -1 & 0 & S \\ \beta & \omega_x & \omega_y & \gamma & \psi \end{vmatrix} \quad (4.1.5)$$

$$B = \begin{vmatrix} 0 \\ -\bar{M}_x^{\delta_\gamma} \cdot K_\psi \\ 0 \\ 0 \\ 0 \end{vmatrix} \quad (4.1.6)$$

Для определения ПФ и $W_\beta^{\Psi_{\text{зад}}}$ находят выражения для определителей Δ и Δ_β , где Δ - главный определитель, а Δ_β - частный определитель (в матрице A столбец, соответствующий β заменён на столбец $\Psi_{\text{зад}}$).

В результате получим:

$$W_\beta^{\Psi_{\text{зад}}} = \frac{\Delta_\beta}{\Delta} = \frac{B_3 s^3 + B_2 s^2 + B_1 s}{s^5 + A_4 s^4 + A_3 s^3 + A_2 s^2 + A_1 s + A_0} \quad (4.1.7)$$

Чтобы выполнялось условие $\beta = 0$, коэффициенты B_1, B_2, B_3 также должны быть равны нулю. Последнее реализовать технически трудно, поэтому на практике ограничиваются требованием отсутствия скольжения только в установившемся развороте, для чего достаточно лишь выполнение условия

$B_1 = 0$, что соответствует

$$K_{\omega_y} = \left(K_{\omega_y} + \frac{\bar{M}_y^{\omega_y}}{\bar{M}_y^{\delta_H}} \right) \frac{g}{V} \quad (4.1.8)$$

Передаточное число K_{ω_y} определяют, исходя из требований к контуру демпфирования изолированного канала РН:

$$K_{\omega_y} = -\frac{\bar{M}_y^{\omega_y}}{\bar{M}_y^{\delta_H}} - \frac{1}{\bar{M}_y^{\delta_H}} \sqrt{-2\bar{M}_y^{\beta} - (\bar{Z}^{\beta})^2} \quad (4.1.9)$$

При выборе передаточного числа $K_{\psi}^{\omega_y}$ используется упрощённая система уравнений самолёта в виде (1.2.6). Так как переменная δ_H в этой системе не фигурирует, то для получения системы уравнений контура стабилизации угла рыскания достаточно к системе (1.2.6) добавить уравнение (4.1.2). В итоге получим систему уравнений, описывающих динамику координированного разворота.

Функциональная схема системы стабилизации угла рыскания при использовании координированного разворота (рисунок 22):

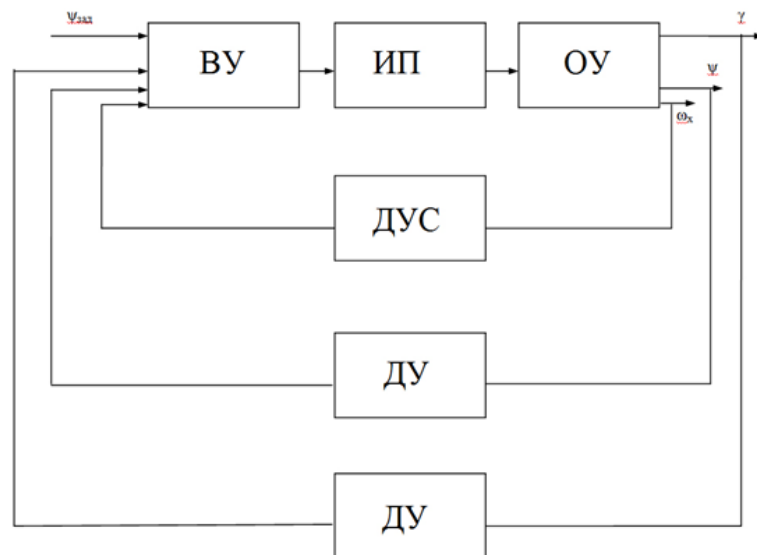


Рисунок 22. Функциональная схема системы стабилизации угла рыскания при использовании координированного разворота

Этой системе соответствует структурная схема, показанная на рисунке 23.

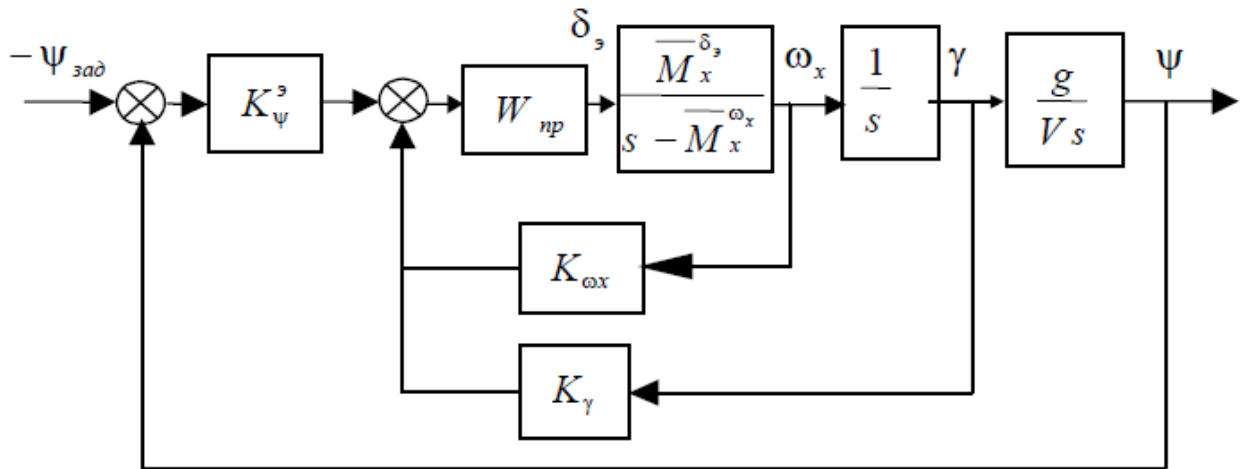


Рисунок 23. Система управления углом рыскания в режиме координированного разворота

Из этой схемы можно получить передаточную функцию замкнутой системы:

$$W_{\psi}^{-\psi_{\text{зад}}}(s) = \frac{-142.8K_{\psi}^3}{s^5 + 29.28s^4 + 428.28s^3 + (2800K_{\omega_x} + 400)s^2 + 2800K_{\gamma}s + 142.8K_{\psi}^3} \quad (4.1.10)$$

4.2. Синтез системы управления углом рыскания в режиме плоского разворота

Коэффициенты характеристического многочлена ПФ 14:

$$\begin{cases} A_5 = 1 \\ A_4 = 29.28 \\ A_3 = 428.28 \\ A_2 = 2800K_{\omega_x} + 400 \\ A_1 = 2800K_{\gamma} \\ A_0 = 142.8K_{\psi}^3 \end{cases} \quad (4.2.1)$$

Сформулируем задачу математического программирования

$$\min(z) = \min(A_1 A_0^{-1}) \quad (4.2.2)$$

При ограничениях

$$\begin{cases} \lambda_1 = A_0^{-1} A_1 A_2 A_3^{-1} \geq 2.15 \\ \lambda_2 = A_1^{-1} A_2 A_3 A_4^{-1} \geq 2.15 \\ \lambda_3 = A_2^{-1} A_3 A_4 A_5^{-1} \geq 2.15 \\ \delta_1 = A_0^{-1} A_1^2 A_2^{-1} \geq 2.1 \\ \delta_2 = A_1^{-1} A_2^2 A_3^{-1} \geq 2.1 \\ \delta_3 = A_2^{-1} A_3^2 A_4^{-1} \geq 2.1 \\ 0.1 \leq K_{\omega_y} \leq 300 \\ 0.1 \leq K_{\gamma} \leq 300 \\ 0.1 \leq K_{\psi}^3 \leq 300 \end{cases} \quad (4.2.3)$$

После определения целевой функции и нелинейных ограничений, применим функцию `fmincon()`. Полученные коэффициенты:

$$K_{\omega_x} = 0.9225 \quad (4.2.4)$$

$$K_{\gamma} = 3.5337 \quad (4.2.5)$$

$$K_{\psi}^3 = 109.4343 \quad (4.2.6)$$

Получили передаточную функцию:

$$W_{\psi}^{-\Psi_{\text{зад}}}(s) = -\frac{15630}{s^5 + 29.28s^4 + 428.3s^3 + 2983s^2 + 9894s + 15630} \quad (4.2.7)$$

Переходный процесс с заданными коэффициентами показан на рисунке 24.

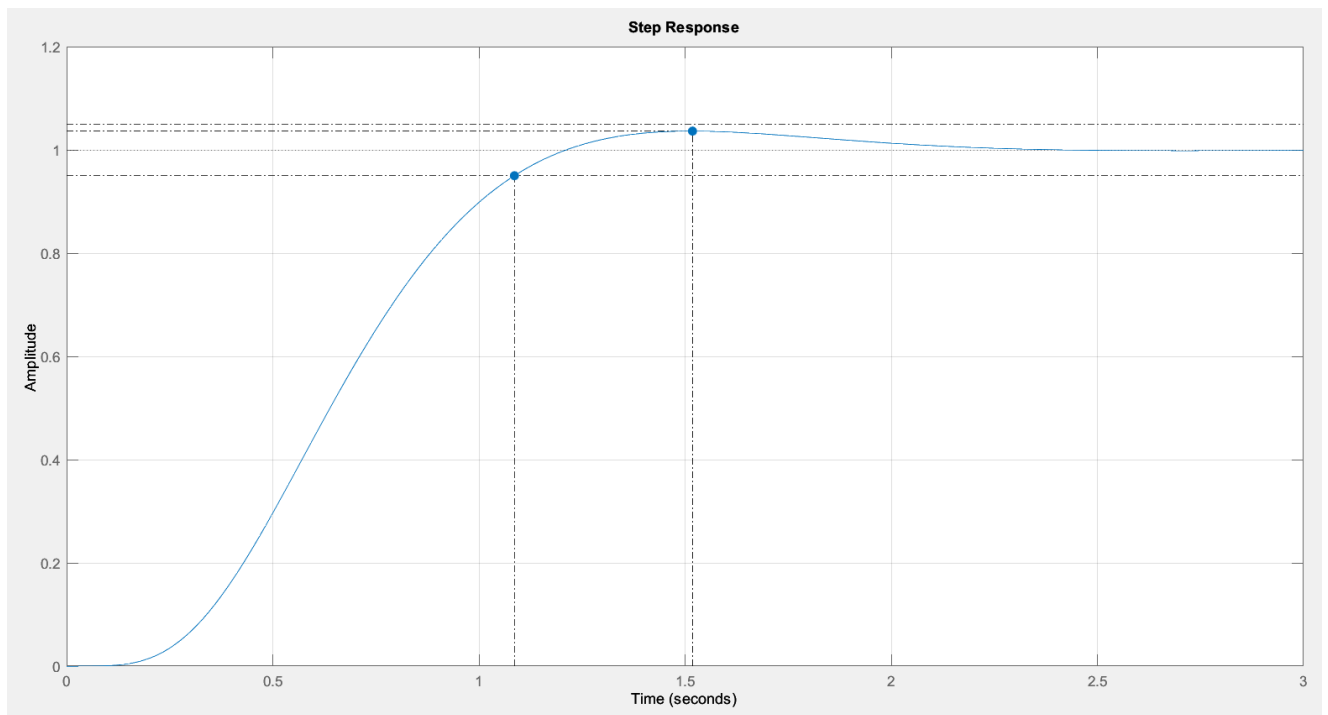


Рисунок 24. Переходный процесс в канале рыскания при координированном развороте
после синтеза функцией `fmicon`

$$t_{п.п.} = 1.09; h_{max} = 1.04; \sigma = 3.63\%$$

4.3. Подбор коэффициентов в NCD-блоке для САУ углом рыскания в режиме координированного разворота

Осуществим подбор коэффициентов в системе Simulink при помощи NCD-блока для с параметрами моделирования.

$$t_{п.п.} = 1; h_{max} = 1.05; \sigma = 5\%$$

Схема моделирования представлена на рисунке 25.

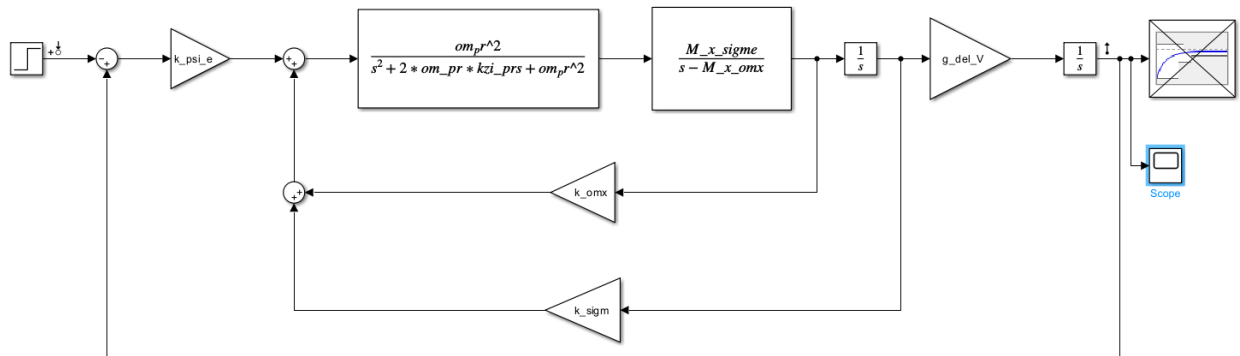


Рисунок 25. Схема моделирования для САУ углом рыскания в режиме координированного разворота

В ходе моделирования получаем коэффициенты

$$K_{\omega_x} = 0.87424 \quad (4.3.1)$$

$$K_{\gamma} = 3.5647 \quad (4.3.2)$$

$$K_{\psi}^3 = 116.03 \quad (4.3.3)$$

Переходный процесс при таких параметрах показан на рисунке 26.

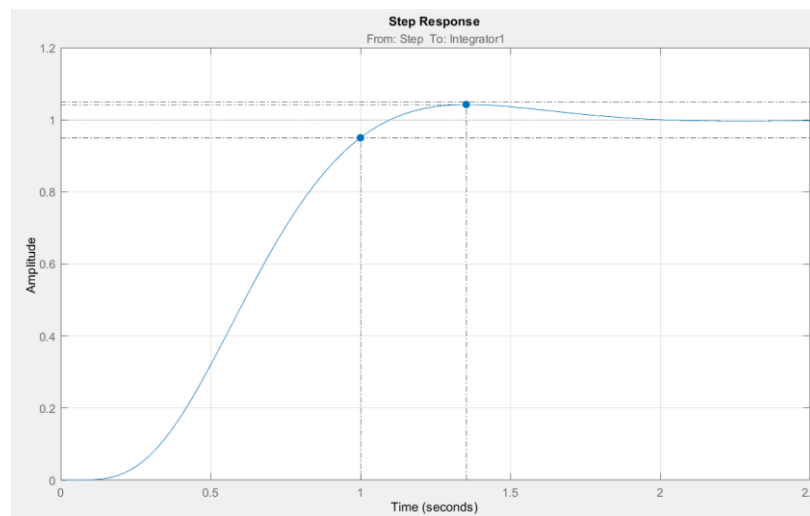


Рисунок 26. Переходный процесс в канале рыскания при координированном развороте после моделирования в NCD-блоке

$$t_{п.п.} = 1; h_{max} = 1.04; \sigma = 4.25\%$$

4.4. Построение области устойчивости для САУ углом рыскания в режиме координированного разворота

Построим область устойчивости в зависимости от параметров K_{ω_y} , K_γ и K_ψ^3 . Для этого воспользуемся критерием Рауса. Для системы пятого порядка условия устойчивости по этому критерию выглядят следующим образом:

$$\begin{cases} A_5 > 0 \\ A_4 > 0 \\ A_3 > 0 \\ A_2 > 0 \\ A_1 > 0 \\ A_0 > 0 \\ (A_1 A_2 - A_0 A_3)(A_3 A_4 - A_2 A_5) - (A_1 A_4 - A_0 A_5)^2 > 0 \end{cases} \quad (4.4.1)$$

Где $A_0, A_1, A_2, A_3, A_4, A_5$ – коэффициенты характеристического многочлена полной системы с учётом привода (система (4.2.1)).

Область устойчивости – это область, которая соответствует выполнению критерия Рауса для системы 5-го порядка в данном случае. Данная область была построена в пространстве 3-х параметров и показана на рисунке 27.

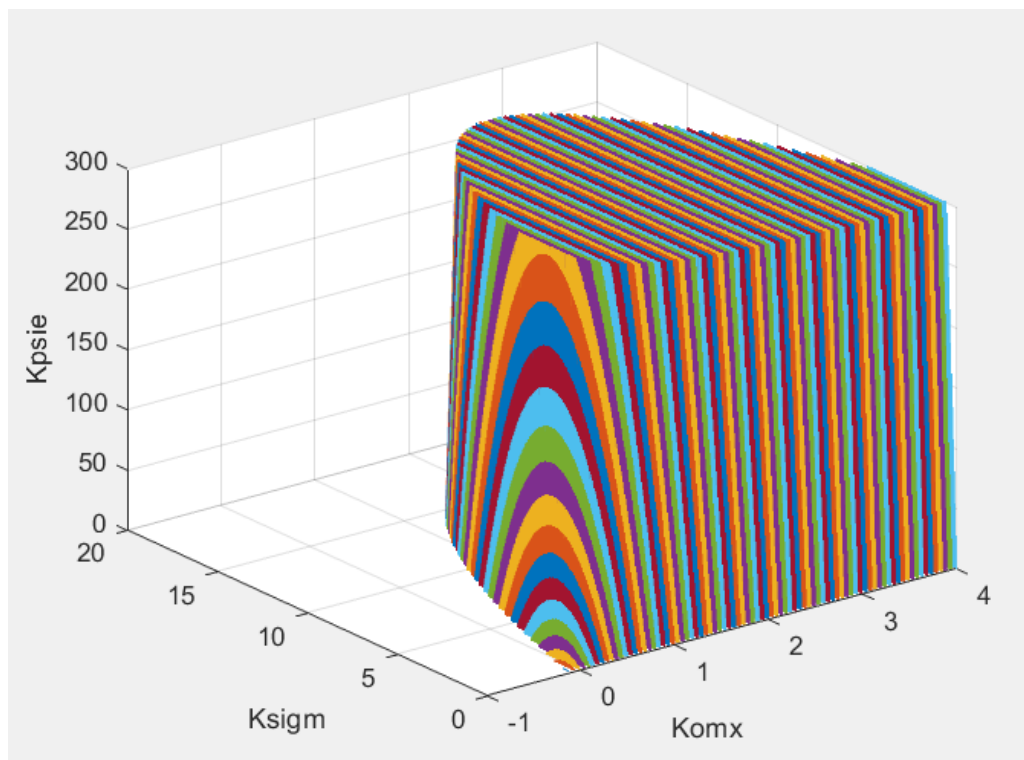


Рисунок 27. Область устойчивости для системы управления рысканием в режиме координированного разворота

4.5. Построение области устойчивости при использовании достаточных условий устойчивости для САУ рыскания при плоском развороте и сравнение такого метода с методом построения по критерию Рауса

Построим область устойчивости в зависимости от параметров K_{ω_y} , K_γ и K_ψ^3 , используя достаточные условия устойчивости. Для системы пятого порядка условия устойчивости по этому критерию выглядят следующим образом:

$$\begin{cases} \lambda_1 = A_0^{-1}A_1A_2A_3^{-1} \geq 2.15 \\ \lambda_2 = A_1^{-1}A_2A_3A_4^{-1} \geq 2.15 \\ \lambda_3 = A_2^{-1}A_3A_4A_5^{-1} \geq 2.15 \end{cases} \quad (3.5.1)$$

Где $A_0, A_1, A_2, A_3, A_4, A_5$ – коэффициенты характеристического многочлена полной системы с учётом привода (система (4.2.1)).

Область устойчивости, удовлетворяющая системе (3.5.1) представлена на рисунке 28.

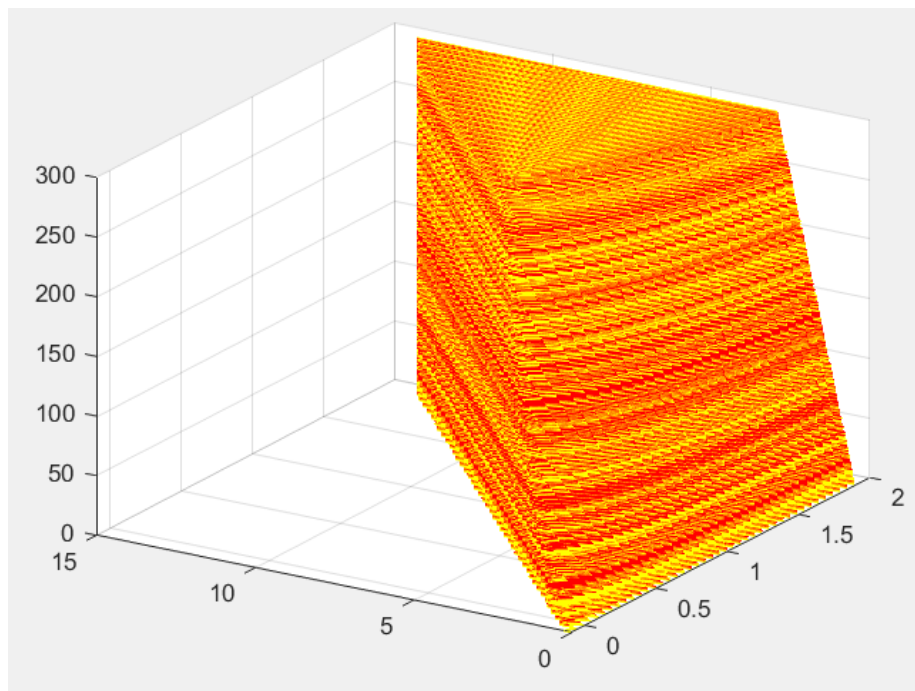


Рисунок 28. Построение области устойчивости по достаточным условиям устойчивости для САУ угла рыскания при координированном развороте

Теперь построим область устойчивости по критерию Рауса, как в главе 4.4, только будем брать сечения плоскостями K_{ω_y} и K_γ и сделаем шаг гораздо больше.

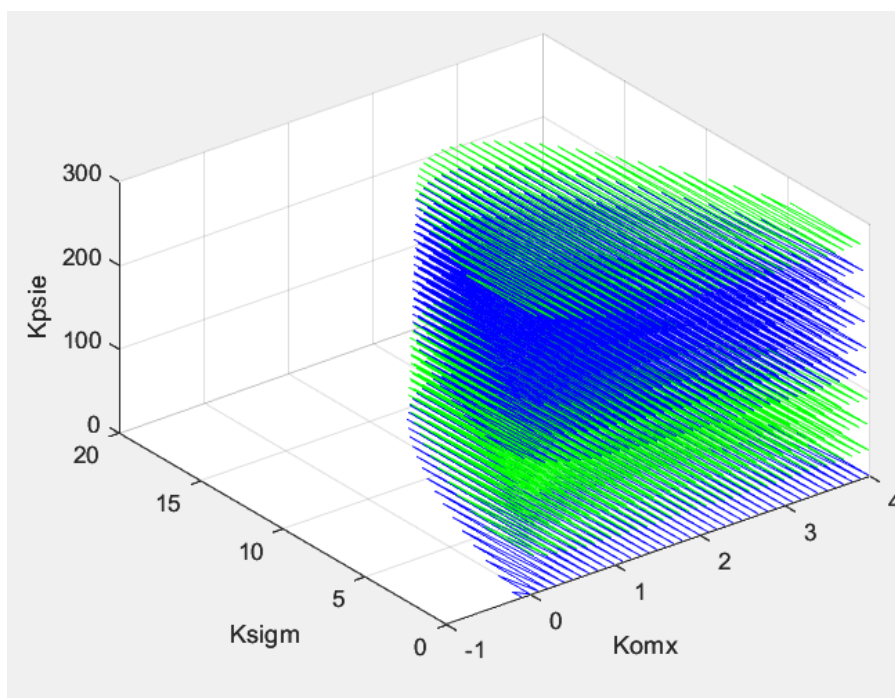


Рисунок 29. Область устойчивости по критерию Рауса

А теперь соединим оба рисунка и получим область по достаточным условиям устойчивости внутри области по критерию Рауса

Вид спереди изображён на рисунке 30.

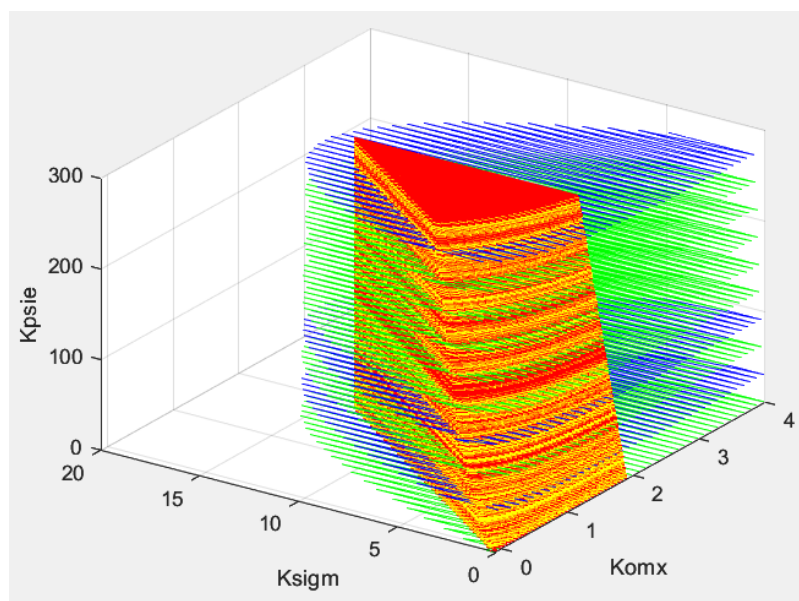


Рисунок 30. Области устойчивости по двум критериям для САУ углом рыскания при координированном развороте (вид спереди)

Вид сзади изображён на рисунке 31.

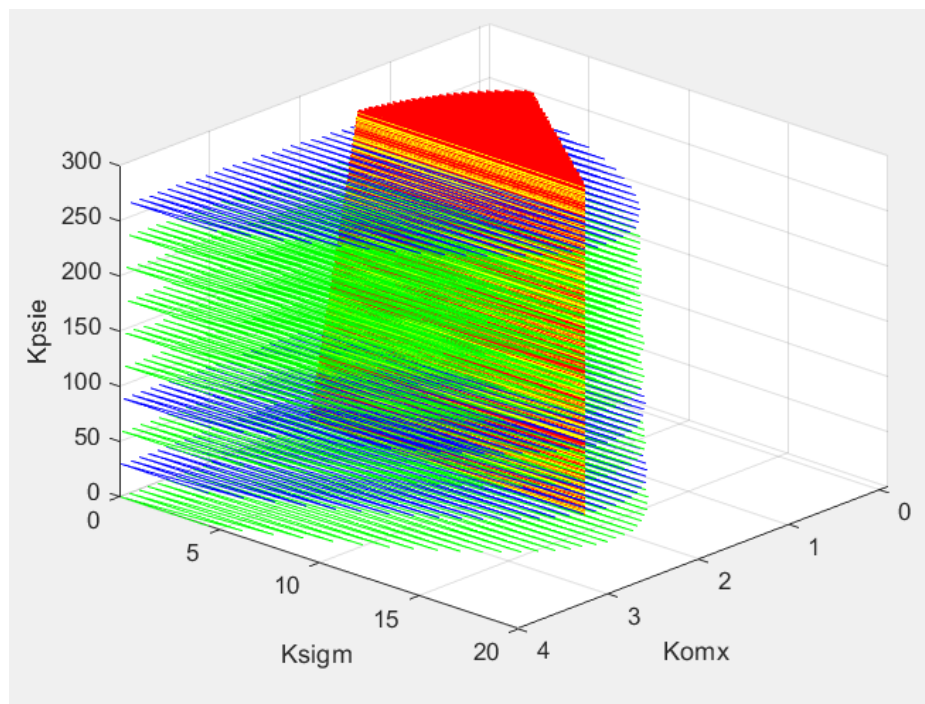


Рисунок 31. Области устойчивости по двум критериям для САУ углом рыскания при координированном развороте (вид сзади)

Видно, что область устойчивости, построенная по критерию достаточного условия устойчивости (синяя), полностью лежит внутри области, построенной по критерию Рауса (красная).

Площадь синей области примерно в 3 раза меньше красной области, кроме того неравенств, по которым строится красная область 7 для данного случая, а неравенств, по которым строится синяя область 3 для данного случая. Это означает, что есть возможность уменьшить скорость построения области по достаточным условиям устойчивости примерно в 7 раз.

Также видно, что большая часть точек синей области лежит условно далеко от границы красной области, а значит, коэффициенты, выбранные внутри синей области почти всегда будут давать качественный переходный процесс.

Построим вид сверху для области, построенной по критерию достаточного условия устойчивости.

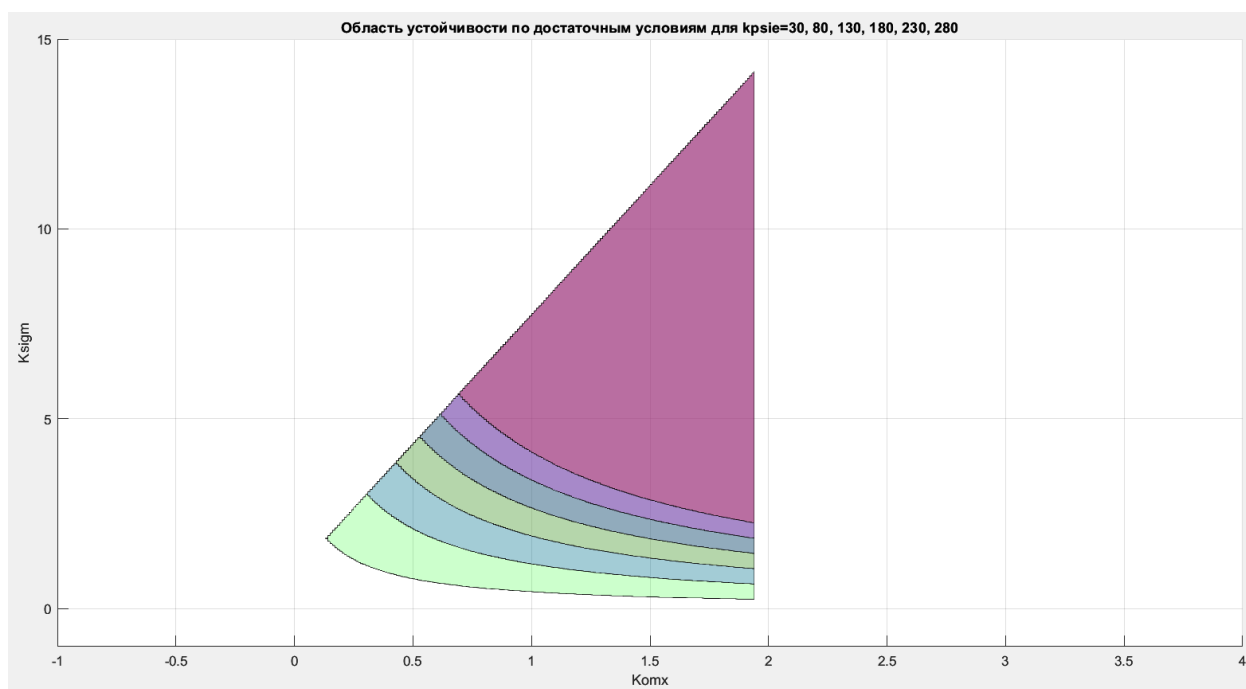


Рисунок 32. Вид сверху на область устойчивости по достаточным условиям устойчивости для САУ углом рыскания при координированном развороте при значении $K_{\psi}^3 = 30, 80, 130, 180, 230, 280$

При этом наибольшая область построена при коэффициенте $K_{\psi}^3 = 30$, а далее с возрастанием коэффициента, уменьшалась область.

Построим вид сверху для области, построенной по критерию достаточного условия устойчивости.

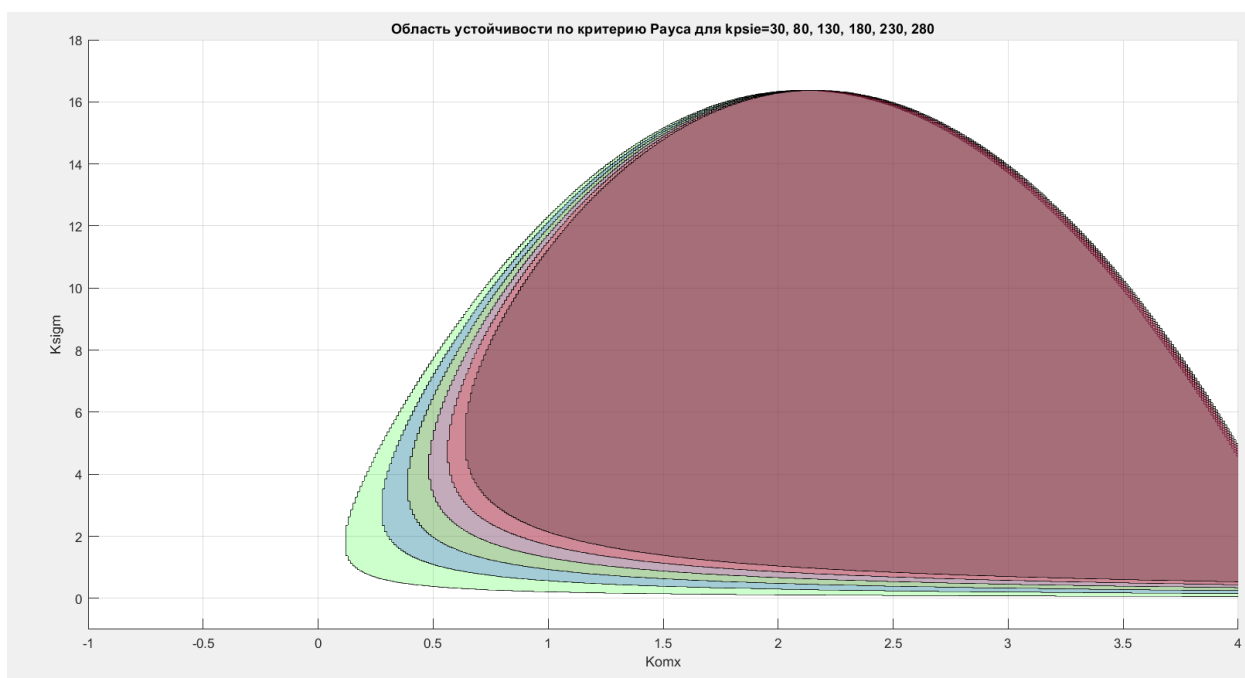


Рисунок 33. Вид сверху на область устойчивости по критерию Рауса для САУ углом рыскания при координированном развороте при значении $K\psi^3 = 30, 80, 130, 180, 230, 280$

При этом наибольшая область построена при коэффициенте $K\psi^3 = 30$, а далее с возрастанием коэффициента, уменьшалась область.

Теперь возьмём какое-то одно значение K_{ψ}^3 , например, в рабочей точке из главы 4.2. Тогда $K_{\psi}^3 = 109.4343$. Построим области устойчивости по 2-м критериям в одной системе координат.

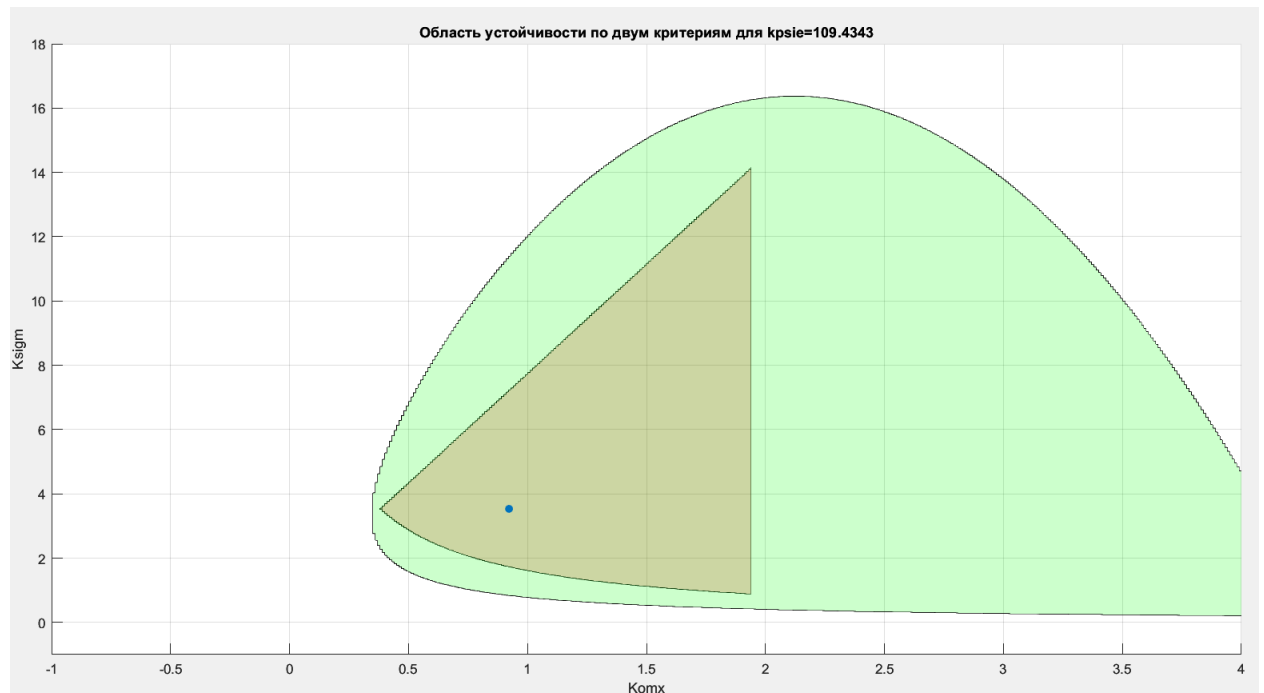


Рисунок 34. Области устойчивости по двум критериям для САУ углом рыскания при плоском развороте для $K_{\psi}^3 = 109.4343$

Видно, что область устойчивости, построенная по критерию достаточного условия устойчивости (красная), полностью лежит внутри области, построенной по критерию Рауса (зелёная).

Площадь красной области примерно в 3.5 раза меньше зелёной области, кроме того неравенств, по которым строится зелёная область 7 для данного случая, а неравенств, по которым строится красная область 3 для данного случая. Это означает, что есть возможность уменьшить скорость построения области по достаточным условиям устойчивости примерно в 8 раз.

Также видно, что большая часть точек синей области лежит условно далеко от границы красной области, а значит, коэффициенты, выбранные внутри синей области почти всегда будут давать качественный переходный процесс.

Заключение

1. Разработана система управления углом крена на указанном режиме полета, удовлетворяющая требованиям к точности, быстродействию и качеству процессов управления (без учета динамики привода $t_{п.п.} = 0.651$; $h_{max} = 1.02$; $\sigma = 1.83\%$, с учетом динамики привода $t_{п.п.} = 0.522$; $h_{max} = 1.01$; $\sigma = 0.727\%$)

Получены следующие значения коэффициентов передачи:

$$K_{\gamma} = 3.6957$$

$$K_{\omega_x} = 1$$

Разработана система управления углом рыскания при использовании плоского разворота на указанном режиме полета, удовлетворяющая требованиям ко времени переходного процесса, но немного нарушающая требование на перерегулирование ($t_{п.п.} = 5.34$; $h_{max} = 1$; $\sigma = 0\%$)

Получены следующие значения коэффициентов передачи:

$$K_{\psi} = 9.0425$$

$$K_{\omega_y} = 2.7468$$

3. Разработана система управления углом рыскания при использовании координированного разворота на указанном режиме полета, удовлетворяющая требованиям к точности, быстродействию и качеству процессов управления ($t_{п.п.} = 1$; $h_{max} = 1$; $\sigma = 0\%$).

Получены следующие значения коэффициентов передачи:

$$K_{\gamma} = 3.5337$$

$$K_{\omega_x} = 0.9225$$

$$K_{\psi}^{\exists} = 109.4343$$

Список использованных источников

1. Лекции по дисциплине «Проектирование систем автоматического управления летательных аппаратов». Лектор: Мулин П.В.

Дата обращения: 01.11.2022.

2. Лекции по дисциплине «Системы автоматического управления воздушными летательными аппаратами». Лектор: Рыбников С.И.

Дата обращения: 02.11.2022.

Приложение 1

```
clc
clear

% Определение параметров системы
H = 5;
M = 0.6;
V = 192;
Zb = -0.2;
sin = 0.08;
cos = 1;
g_del_V = 0.051;

M_x_b = -5.8;
M_x_omx = -1;
M_x_omy = -0.2;
M_x_sigme = -7;
M_y_b = -3;
M_y_omx = -0.05;
M_y_omy = -0.2;
M_y_sigmn = -2.5;

om_pr = 20;
kzi_pr = sqrt(2)/2;

% Синтез параметров
a = 4; % Коэффициент пропорциональности между tpr и временем релаксации
kzi = sqrt(a^2/(pi^2+a^2)); % Коэффициент затухания
t_pp = 1; % Задаём примерное время переходного процесса
om0 = a / kzi / t_pp; % Определяем собственную частоту колебаний
T = 2*pi / (om0*sqrt(1-kzi^2)); % Определяем время релаксации

disp('Синтезированные коэффициенты')
k_sigme = - om0^2 / M_x_sigme % Значение коэффициент k_sigme
k_omx = - (M_x_omx + 2*kzi*om0) / M_x_sigme % Определяем коэффициент k_omx

% Построение переходного процесса в системе с учётом привода
% с синтезированными коэффициентами
W1 = tf(M_x_sigme, [1 -M_x_omx]); % ПФ по угловой скорости
W2 = tf(om_pr^2, [1 2*kzi_pr*om_pr om_pr^2]); % ПФ привода
W3 = W1*W2; % Связываем W1 и W2
integr = tf(1, [1, 0]); % ПФ интегратора
W4 = feedback(W3, k_omx, 1); % Положительная ОС по угловой скорости
W5 = k_sigme * W4 * integr; % Разомкнутая цепь по углу крена
disp('ПФ замкнутой цепи по углу крена с учётом привода')
W6 = feedback(W5, 1, 1) % Замкнутая цепь по углу крена
disp('Корни характеристического уравнения')
roots(W6.denominator{1}) % Корни характеристического уравнения
figure;
step(-1*W6); % Построение переходного процесса

% Построение переходного процесса в системе без учёта привода
% с синтезированными коэффициентами
W1 = tf(M_x_sigme, [1 -M_x_omx]); % ПФ по угловой скорости
integr = tf(1, [1, 0]); % ПФ интегратора
```

```

W2 = feedback(W1, k_omx, 1); % Положительная ОС по угловой скорости
W3 = k_sigm * W2 * integr; % Разомкнутая цепь по углу крена
disp('ПФ замкнутой цепи по углу крена без учёта привода')
W4 = feedback(W3, 1, 1) % Замкнутая цепь по углу крена
disp('Корни характеристического уравнения')
roots(W4.denominator{1}) % Корни характеристического уравнения
hold on
step(-1*W4); % Построение переходного процесса
hold off
grid on
legend('С учётом динамики привода', 'Без учёта динамики привода')

% Корни, полученные при моделировании в NCD-блоке
disp('Полученные в NCD-блоке коэффициенты')
k_sigm = 5.4231
k_omx = 1.2597

```

```

% Те же самые построение только уже для подобранных в NCD-блоке
% коэффициентов
W1 = tf(M_x_signe, [1 -M_x_omx]);
W2 = tf(om_pr^2, [1 2*kzi_pr*om_pr om_pr^2]);
W3 = W1*W2;
integr = tf(1, [1, 0]);
W4 = feedback(W3, k_omx, 1);
W5 = k_sigm * W4 * integr;
disp('ПФ замкнутой цепи по углу крена с учётом привода')
W6 = feedback(W5, 1, 1)
disp('Корни характеристического уравнения')
roots(W6.denominator{1})
figure;
step(-1*W6)

```

```

W1 = tf(M_x_signe, [1 -M_x_omx]);
integr = tf(1, [1, 0]);
W2 = feedback(W1, k_omx, 1);
W3 = k_sigm * W2 * integr;
disp('ПФ замкнутой цепи по углу крена без учёта привода')
W4 = feedback(W3, 1, 1)
disp('Корни характеристического уравнения')
roots(W4.denominator{1})
hold on
step(-1*W4)
hold off
legend('С учётом динамики привода', 'Без учёта динамики привода')
grid on

```

Программа 1. Основной код для системы управления стабилизацией крена

```

clc
clear

syms k_sigm k_omx s

% Определение параметров системы
H = 5;
M = 0.6;
V = 192;
Zb = -0.2;

```

```

sin = 0.08;
cos = 1;
g_del_V = 0.051;

M_x_b = -5.8;
M_x_omx = -1;
M_x_omy = -0.2;
M_x_sigme = -7;
M_y_b = -3;
M_y_omx = -0.05;
M_y_omy = -0.2;
M_y_sigmn = -2.5;

om_pr = 20;
kzi_pr = sqrt(2)/2;

W_pr = om_pr^2 / (s^2 + 2*om_pr*kzi_pr*s + om_pr^2); % ПФ привода
W1 = collect(M_x_sigme / (s - M_x_omx)); % ПФ по угловой скорости
W2 = collect(W1 * W_pr); % Связываем W1 и W2
W3 = collect(W2 / (1 - W2*k_omx)); % Положительная ОС по угловой скорости
W4 = collect(k_sigmn * W3 / s); % Разомкнутая цепь по углу крена
W5 = collect(W4 / (1 - W4)); % Замкнутая цепь по углу крена
disp('ПФ замкнутой цепи по углу крена с учётом привода')
pretty(W5)

[num, den] = numden(W5);
disp('Числитель передаточной функции')
num = vpa(collect(num), 4)
disp('Знаменатель передаточной функции')
den = vpa(collect(den), 4)

% Выделим коэффициенты в числителе и знаменателе
% Функция coeffs запишет их от младшей степени к старшей
koefs_num = coeffs(num, 's'); % Массив коэффициентов числителя
koefs_den = coeffs(den, 's'); % Массив коэффициентов знаменателя

% Перевернём массивы на 180 градусов, чтобы получить коэффициенты
% от старшего ко младшему
koefs_num = rot90(rot90(koefs_num));
koefs_den = rot90(rot90(koefs_den));

% Запишем соответствующие коэффициенты в соответствующие переменные
B0 = koefs_num(1);
A4 = koefs_den(1);
A3 = koefs_den(2);
A2 = koefs_den(3);
A1 = koefs_den(4);
A0 = koefs_den(5);

% Запишем неравенства для критерия Рауса
eq1 = A0 > 0;
eq2 = A1 > 0;
eq3 = A2 > 0;
eq4 = A3 > 0;
eq5 = A4 > 0;
eq6 = simplify(A1*A2*A3 - A1^2*A4 - A0*A3^2) > 0;

```



```

% Преобразуем неравенства из символьного типа данных в функции
eq1 = matlabFunction(eq1);
eq2 = matlabFunction(eq2);
eq3 = matlabFunction(eq3);
eq4 = matlabFunction(eq4);
eq5 = matlabFunction(eq5);
eq6 = matlabFunction(eq6);

% Задание сетки
l1 = [-2:0.005:5];
l2 = [-2:0.005:20];
[ko, ks] = meshgrid(l1, l2);

% Проверка устойчивости узлов сетки
c1 = eq1(ks);
c2 = eq2(ko);
c3 = eq3();
c4 = eq4();
c5 = eq5();
c6 = eq6(ko, ks);

figure;
contourf(l1, l2, c1 & c2 & c3 & c4 & c5 & c6, [1 1 1 1 1 1])
colormap lines
grid on
hold on
% Синтезированные коэффициенты
k_omx = 1;
k_sigm = 3.6957;
pnt1 = scatter(k_omx, k_sigm, 'r','filled');
% Подобранные коэффициенты
k_omx = 1.2597;
k_sigm = 5.4231;
pnt2 = scatter(k_omx, k_sigm, 'b','filled');
legend([pnt1, pnt2], "Синтезированные коэффициенты", "Подобранные коэффициенты")
hold off
xlabel('Komx')
ylabel('Ksigm')

```

Программа 2. Область устойчивости для системы управления стабилизацией крена

```

clc
clear

syms k_sigm k_omx s

% Определение параметров системы
H = 5;
M = 0.6;
V = 192;
Zb = -0.2;
sin = 0.08;
cos = 1;
g_del_V = 0.051;

M_x_b = -5.8;
M_x_omx = -1;

```

```

M_x_omy = -0.2;
M_x_sigme = -7;
M_y_b = -3;
M_y_omx = -0.05;
M_y_omy = -0.2;
M_y_sigmn = -2.5;

om_pr = 20;
kzi_pr = sqrt(2)/2;

W_pr = om_pr^2 / (s^2 + 2*om_pr*kzi_pr*s + om_pr^2); % ПФ привода
W1 = collect(M_x_sigme / (s - M_x_omx)); % ПФ по угловой скорости
W2 = collect(W1 * W_pr); % Связываем W1 и W2
W3 = collect(W2 / (1 - W2*k_omx)); % Положительная ОС по угловой скорости
W4 = collect(k_sigm * W3 / s); % Разомкнутая цепь по углу крена
W5 = collect(W4 / (1 - W4)); % Замкнутая цепь по углу крена
disp('ПФ замкнутой цепи по углу крена с учётом привода')
pretty(W5)

[num, den] = numden(W5);
disp('Числитель передаточной функции')
num = vpa(collect(num), 4)
disp('Знаменатель передаточной функции')
den = vpa(collect(den), 4)

% Выделим коэффициенты в числителе и знаменателе
% Функция coeffs запишет их от младшей степени к старшей
koefs_num = coeffs(num, 's'); % Массив коэффициентов числителя
koefs_den = coeffs(den, 's'); % Массив коэффициентов знаменателя

% Перевернём массивы на 180 градусов, чтобы получить коэффициенты
% от старшего ко младшему
koefs_num = rot90(rot90(koefs_num));
koefs_den = rot90(rot90(koefs_den));

% Запишем соответствующие коэффициенты в соответствующие переменные
B0 = koefs_num(1);
A4 = koefs_den(1);
A3 = koefs_den(2);
A2 = koefs_den(3);
A1 = koefs_den(4);
A0 = koefs_den(5);

% Запишем неравенства для достаточных условий устойчивости через лямбды
lambd_star = 2.15;
eq1d = A0^(-1) * A1 * A2 * A3^(-1) >= lambd_star;
eq2d = A1^(-1) * A2 * A3 * A4^(-1) >= lambd_star;

% Преобразуем неравенства из символьного типа данных в функции
eq1d = matlabFunction(eq1d);
eq2d = matlabFunction(eq2d);

% Запишем неравенства для критерия Рауса
eq1r = A0 > 0;
eq2r = A1 > 0;

```

```

eq3r = A2 > 0;
eq4r = A3 > 0;
eq5r = A4 > 0;
eq6r = simplify(A1*A2*A3 - A1^2*A4 - A0*A3^2) > 0;

% Преобразуем неравенства из символьного типа данных в функции
eq1r = matlabFunction(eq1r);
eq2r = matlabFunction(eq2r);
eq3r = matlabFunction(eq3r);
eq4r = matlabFunction(eq4r);
eq5r = matlabFunction(eq5r);
eq6r = matlabFunction(eq6r);

% Задание сетки
l1 = [-2:0.005:5];
l2 = [-2:0.005:20];
[ko, ks] = meshgrid(l1, l2);

% Проверка устойчивости узлов сетки на достаточные условия устойчивости
c1d = eq1d(ko, ks);
c2d = eq2d(ko);

% Проверка устойчивости узлов сетки по критерию Рауса
c1r = eq1r(ks);
c2r = eq2r(ko);
c3r = eq3r();
c4r = eq4r();
c5r = eq5r();
c6r = eq6r(ko, ks);

figure;
grid on
hold on
% Контур для области устойчивости по критерию Рауса
contourf(l1, l2, c1r & c2r & c3r & c4r & c5r & c6r, [1 1 1 1 1 1], FaceAlpha=0.5,
FaceColor="red")
% Контур для области устойчивости по достаточным условиям устойчивости
contourf(l1, l2, c1d & c2d, [1 1], FaceAlpha=0.5, FaceColor="blue")
title('Красная область - достаточные условия устойчивости. Синяя область - критерий Рауса')

% Синтезированные коэффициенты
k_omx = 1;
k_sigm = 3.6957;
pnt1 = scatter(k_omx, k_sigm, 'y','filled');
% Подобранные коэффициенты
k_omx = 1.2597;
k_sigm = 5.4231;
pnt2 = scatter(k_omx, k_sigm, 'b','filled');
legend([pnt1, pnt2], "Синтезированные коэффициенты", "Подобранные коэффициенты")
xlabel('Комх')
ylabel('Ksigm')

```

Программа 3. Программа для областей устойчивости для САУ крена для задания на зачёт

Приложение 2

```
clc
clear

syms k_psi k_omegay s

% Определение параметров системы
H = 5;
M = 0.6;
V = 192;
Zb = -0.2;
sin = 0.08;
cos = 1;
g_del_V = 0.051;

M_x_b = -5.8;
M_x_omx = -1;
M_x_omy = -0.2;
M_x_sigme = -7;
M_y_b = -3;
M_y_omx = -0.05;
M_y_omy = -0.2;
M_y_sigmn = -2.5;

om_pr = 20;
kzi_pr = sqrt(2)/2;

% Передаточная функция привода
W_pr = om_pr^2 / (s^2 + 2*om_pr*kzi_pr*s + om_pr^2);
% Передаточная функция по угловой скорости рысканья с учётом привода
W_sigmn_omy_raz = collect((M_y_sigmn*s + (-M_y_sigmn*Zb)) / (s^2 + (-M_y_omy-Zb)*s +
(M_y_omy*Zb-M_y_b)) * W_pr);
% Обратная связь с коэффициентом обратной связи k_omegay
W_sigmn_omy_zamk = collect(W_sigmn_omy_raz / (1 - k_omegay*W_sigmn_omy_raz));
% Разомкнутая передаточная функция по углу рысканья с замкнутым внутренним
% контуром
W_pszad_psi_raz = collect(k_psi * W_sigmn_omy_zamk / s);
% Передаточная функция замкнутой системы по углу рысканья
W_pszad_psi_zamk = collect(W_pszad_psi_raz / (1 - W_pszad_psi_raz));

% Выделим числитель и знаменатель итоговой ПФ
[num, den] = numden(W_pszad_psi_zamk);
num = collect(num);
den = collect(den);
% Выделим коэффициенты числителя и знаменателя
% При этом сами коэффициенты будут записаны в порядке от младшего к
% старшему, то есть перевернутся
koefs_num = coeffs(num, 's');
koefs_den = coeffs(den, 's');
% Выделим старший коэффициент знаменателя. В списке он будет последний
starsh_koef = koefs_den(length(koefs_den));
% Поделим все коэффициенты на это число
koefs_num = vpa(koefs_num / starsh_koef);
koefs_den = vpa(koefs_den / starsh_koef);
% Развернём оба списка с коэффициентами, чтобы они шли от старшего ко
% младшему
koefs_num = rot90(rot90(koefs_num));
koefs_den = rot90(rot90(koefs_den));
```

```

% Преобразованные коэффициенты передаточной функции
B1 = koefs_num(1);
B0 = koefs_num(2);
A5 = koefs_den(1);
A4 = koefs_den(2);
A3 = koefs_den(3);
A2 = koefs_den(4);
A1 = koefs_den(5);
A0 = koefs_den(6);

% Передаточная функция в нормальном виде (старший коэффициент знаменателя
% равен 1
num = (B1*s + B0);
den = (A5*s^5 + A4*s^4 + A3*s^3 + A2*s^2 + A1*s + A0);
PF = (num / den);
disp('Числитель ПФ')
pretty(num)
disp('Знаменатель ПФ')
pretty(den)

% Зададим параметры для оптимизации в fmincon
k0 = [10, 10];
A = [];
b = [];
Aeg = [];
beg = [];
lb = 0.1*ones(2,1);
ub = 300*ones(2,1);

% Осуществим параметрическую оптимизацию
[x, fval] = fmincon('fun_min', k0, A, b, Aeg, beg, lb, ub, 'nonclon');

% Полученные коэффициенты
k_omegay = x(1)
k_psi = x(2)

% Преобразовывает коэффициенты характеристического полинома,
% подставляя в них найденные значения
B_1 = vpa(-1000.0*x(2));
B_0 = vpa(-200.0*x(2));
A_5 = vpa(1);
A_4 = vpa(28.6842);
A_3 = vpa(414.3537);
A_2 = vpa(1000*x(1) + 245.98);
A_1 = vpa(1000*x(2) + 200*x(1) + 1216);
A_0 = vpa(200.0.*x(2));
num = B_1*s + B_0;
den = A_5*s^5 + A_4*s^4 + A_3*s^3 + A_2*s^2 + A_1*s + A_0;
num = sym2poly(num);
den = sym2poly(den);

% Получаем передаточную функцию и строим переходный процесс
disp('Передаточная функция с полученными коэффициентами')
W_sys = tf(num, den)
figure;
step(-1*W_sys)
grid on

```

Программа 4. Основной код для системы управление рысканием в режиме плоского разворота

```

function f = fun_min(x)
% x(1) = k_omegay, x(2) = k_psi

    A_1 = 1000*x(2) + 200*x(1) + 1216;
    A_0 = 200.0*x(2);

    f = A_1 * A_0^(-1);

end

```

Программа 5. Целевая функция для системы управления углом рыскания в режиме
плоского разворота

```

function [c, seq] = nonclon(x)
% x(1) = k_omegay, x(2) = k_psi

    A_5 = 1;
    A_4 = 28.6842;
    A_3 = 414.3537;
    A_2 = 1000*x(1) + 245.98;
    A_1 = 1000*x(2) + 200*x(1) + 1216;
    A_0 = 200.0*x(2);

    lambd_star = 2.15; % Желаемое значение лямбды
    lambd1 = lambd_star - (A_0^(-1) * A_1 * A_2 * A_3^(-1));
    lambd2 = lambd_star - (A_1^(-1) * A_2 * A_3 * A_4^(-1));
    lambd3 = lambd_star - (A_2^(-1) * A_3 * A_4 * A_5^(-1));

    sigm_star = 2; % Желаемое значение сигмы
    sigm1 = sigm_star - (A_0^(-1) * A_1^2 * A_2^(-1));
    sigm2 = sigm_star - (A_1^(-1) * A_2^2 * A_3^(-1));
    sigm3 = sigm_star - (A_2^(-1) * A_3^2 * A_4^(-1));

    c = [lambd1; lambd2; lambd3; sigm1; sigm2; sigm3];
    seq = [];

end

```

Программа 6. Ограничения для системы управления углом рыскания в режиме плоского
разворота

```

clc
clear

syms k_psi k_omegay s

% Задаём параметры системы
H = 5;
M = 0.6;
V = 192;
Zb = -0.2;
sin = 0.08;
cos = 1;
g_del_V = 0.051;

M_x_b = -5.8;
M_x_omx = -1;
M_x_omy = -0.2;
M_x_sigme = -7;

```

```

M_y_b = -3;
M_y_omx = -0.05;
M_y_omy = -0.2;
M_y_sigmn = -2.5;

om_pr = 20;
kzi_pr = 0.707;

% Передаточная функция привода
W_pr = om_pr^2 / (s^2 + 2*om_pr*kzi_pr*s + om_pr^2);
% Передаточная функция по угловой скорости рысканья с учётом привода
W_sigmn_omy_raz = collect((M_y_sigmn*s + (-M_y_sigmn*Zb)) / (s^2 + (-M_y_omy-Zb)*s +
(M_y_omy*Zb-M_y_b)) * W_pr);
% Обратная связь с коэффициентом обратной связи k_omegay
W_sigmn_omy_zamk = collect(W_sigmn_omy_raz / (1 - k_omegay*W_sigmn_omy_raz));
% Разомкнутая передаточная функция по углу рысканья с замкнутым внутренним
% контуром
W_psi_raz = collect(k_psi * W_sigmn_omy_zamk / s);
% Передаточная функция замкнутой системы по углу рысканья
W_psi_raz_zamk = collect(W_psi_raz / (1 - W_psi_raz));

% Выделим числитель и знаменатель итоговой ПФ
[num, den] = numden(W_psi_raz_zamk);
num = collect(num);
den = collect(den);
% Выделим коэффициенты числителя и знаменателя
% При этом сами коэффициенты будут записаны в порядке от младшего к
% старшему, то есть перевернутся
koefs_num = coeffs(num, 's');
koefs_den = coeffs(den, 's');
% Выделим старший коэффициент знаменателя. В списке он будет последний
starsh_koef = koefs_den(length(koefs_den));
% Поделим все коэффициенты на это число
koefs_num = vpa(koefs_num / starsh_koef);
koefs_den = vpa(koefs_den / starsh_koef);
% Развернём оба списка с коэффициентами, чтобы они шли от старшего ко
% младшему
koefs_num = rot90(rot90(koefs_num));
koefs_den = rot90(rot90(koefs_den));

% Преобразованные коэффициенты передаточной функции
B1 = koefs_num(1);
B0 = koefs_num(2);
A5 = koefs_den(1);
A4 = koefs_den(2);
A3 = koefs_den(3);
A2 = koefs_den(4);
A1 = koefs_den(5);
A0 = koefs_den(6);

% Передаточная функция в нормальном виде (старший коэффициент знаменателя
% равен 1
num = (B1*s + B0);
den = (A5*s^5 + A4*s^4 + A3*s^3 + A2*s^2 + A1*s + A0);
PF = (num / den);
disp('Числитель ПФ')
pretty(num)
disp('Знаменатель ПФ')
pretty(den)

% Запишем неравенства ждя критерия Рауса
eq1 = A0 > 0;

```

```

eq2 = A1 > 0;
eq3 = A2 > 0;
eq4 = A3 > 0;
eq5 = A4 > 0;
eq6 = A5 > 0;
eq7 = simplify((A1*A2-A0*A3)*(A3*A4-A2*A5) - (A1*A4-A0*A5)^2) > 0;

% Преобразуем неравенства из символьного типа данных в функции
eq1 = matlabFunction(eq1);
eq2 = matlabFunction(eq2);
eq3 = matlabFunction(eq3);
eq4 = matlabFunction(eq4);
eq5 = matlabFunction(eq5);
eq6 = matlabFunction(eq6);
eq7 = matlabFunction(eq7);

% Задание сетки
l1 = [-2:0.05:50];
l2 = [-2:0.05:50];
[k0, kp] = meshgrid(l1, l2);

% Проверка устойчивости узлов сетки
c1 = eq1(kp);
c2 = eq2(kp, k0);
c3 = eq3(k0);
c4 = eq4();
c5 = eq5();
c6 = eq6();
c7 = eq7(kp, k0);

% Построение области устойчивости
figure;
contourf(l1, l2, c1 & c2 & c3 & c4 & c5 & c6 & c7, [1 1 1 1 1 1 1])
colormap lines
hold on
% Синтезированные коэффициенты
k_omegay = 2.7468;
k_psi = 9.0425;
pnt1 = scatter(k_omegay, k_psi, 'r', 'filled');
% Подобранные коэффициенты
k_omegay = 2.8172;
k_psi = 9.0425;
pnt2 = scatter(k_omegay, k_psi, 'b', 'filled');
legend([pnt1, pnt2], "Синтезированные коэффициенты", "Подобранные коэффициенты")
hold off
xlabel("Komy")
ylabel("Kpsi")
grid on

```

Программа 7. Построение области устойчивости для системы управления углом рыскания
в режиме плоского разворота

```

clc
clear

syms k_psi k_omegay s

% Задаём параметры системы
H = 5;
M = 0.6;

```



```

V = 192;
Zb = -0.2;
sin = 0.08;
cos = 1;
g_del_V = 0.051;

M_x_b = -5.8;
M_x_omx = -1;
M_x_omy = -0.2;
M_x_sigme = -7;
M_y_b = -3;
M_y_omx = -0.05;
M_y_omy = -0.2;
M_y_sigmn = -2.5;

om_pr = 20;
kzi_pr = 0.707;

% Передаточная функция привода
W_pr = om_pr^2 / (s^2 + 2*om_pr*kzi_pr*s + om_pr^2);
% Передаточная функция по угловой скорости рысканья с учётом привода
W_sigmn_omy_raz = collect((M_y_sigmn*s + (-M_y_sigmn*Zb)) / (s^2 + (-M_y_omy-Zb)*s +
(M_y_omy*Zb-M_y_b)) * W_pr);
% Обратная связь с коэффициентом обратной связи k_omegay
W_sigmn_omy_zamk = collect(W_sigmn_omy_raz / (1 - k_omegay*W_sigmn_omy_raz));
% Разомкнутая передаточная функция по углу рысканья с замкнутым внутренним
% контуром
W_pszad_psi_raz = collect(k_psi * W_sigmn_omy_zamk / s);
% Передаточная функция замкнутой системы по углу рысканья
W_pszad_psi_zamk = collect(W_pszad_psi_raz / (1 - W_pszad_psi_raz));

% Выделим числитель и знаменатель итоговой ПФ
[num, den] = numden(W_pszad_psi_zamk);
num = collect(num);
den = collect(den);
% Выделим коэффициенты числителя и знаменателя
% При этом сами коэффициенты будут записаны в порядке от младшего к
% старшему, то есть перевернутся
koefs_num = coeffs(num, 's');
koefs_den = coeffs(den, 's');
% Выделим старший коэффициент знаменателя. В списке он будет последний
starsh_koef = koefs_den(length(koefs_den));
% Поделим все коэффициенты на это число
koefs_num = vpa(koefs_num / starsh_koef);
koefs_den = vpa(koefs_den / starsh_koef);
% Развернём оба списка с коэффициентами, чтобы они шли от старшего ко
% младшему
koefs_num = rot90(rot90(koefs_num));
koefs_den = rot90(rot90(koefs_den));

% Преобразованные коэффициенты передаточной функции
B1 = koefs_num(1);
B0 = koefs_num(2);
A5 = koefs_den(1);
A4 = koefs_den(2);
A3 = koefs_den(3);
A2 = koefs_den(4);
A1 = koefs_den(5);
A0 = koefs_den(6);

```

```

% Запишем неравенства для достаточных условий устойчивости через лямбды
lambd_star = 2.15;
eq1d = A0^(-1) * A1 * A2 * A3^(-1) >= lambd_star;
eq2d = A1^(-1) * A2 * A3 * A4^(-1) >= lambd_star;
eq3d = A2^(-1) * A3 * A4 * A5^(-1) >= lambd_star;

% Преобразуем неравенства из символьного типа данных в функции
eq1d = matlabFunction(eq1d);
eq2d = matlabFunction(eq2d);
eq3d = matlabFunction(eq3d);

% Запишем неравенства для критерия Рауса
eq1r = A0 > 0;
eq2r = A1 > 0;
eq3r = A2 > 0;
eq4r = A3 > 0;
eq5r = A4 > 0;
eq6r = A5 > 0;
eq7r = simplify((A1*A2-A0*A3)*(A3*A4-A2*A5) - (A1*A4-A0*A5)^2) > 0;

% Преобразуем неравенства из символьного типа данных в функции
eq1r = matlabFunction(eq1r);
eq2r = matlabFunction(eq2r);
eq3r = matlabFunction(eq3r);
eq4r = matlabFunction(eq4r);
eq5r = matlabFunction(eq5r);
eq6r = matlabFunction(eq6r);
eq7r = matlabFunction(eq7r);

% Задание сетки
l1 = [-2:0.01:20];
l2 = [-2:0.01:45];
[k0, kp] = meshgrid(l1, l2);

% Проверка устойчивости узлов сетки на достаточные условия устойчивости
c1d = eq1d(kp, k0);
c2d = eq2d(kp, k0);
c3d = eq3d(k0);

% Проверка устойчивости узлов сетки по критерию Рауса
c1r = eq1r(kp);
c2r = eq2r(kp, k0);
c3r = eq3r(k0);
c4r = eq4r();
c5r = eq5r();
c6r = eq6r();
c7r = eq7r(kp, k0);

figure;
grid on
hold on
% Контур для области устойчивости по критерию Рауса
contourf(l1, l2, c1r & c2r & c3r & c4r & c5r & c6r & c7r, [1 1 1 1 1 1 1],
FaceAlpha=0.5, FaceColor="red")
% Контур для области устойчивости по достаточным условиям устойчивости
contourf(l1, l2, c1d & c2d & c3d, [1 1 1], FaceAlpha=0.5, FaceColor="blue")
title('Красная область - достаточные условия устойчивости. Синяя область - критерий Рауса')

% Синтезированные коэффициенты

```

```

k_omegay = 2.7468;
k_psi = 9.0425;
pnt1 = scatter(k_omegay, k_psi, 'r', 'filled');
% Подобранные коэффициенты
k_omegay = 2.8172;
k_psi = 9.0425;
pnt2 = scatter(k_omegay, k_psi, 'y', 'filled');
legend([pnt1, pnt2], "Синтезированные коэффициенты", "Подобранные коэффициенты")
hold off
xlabel("Komy")
ylabel("Kpsi")
grid on

```

Программа 8. Программа для областей устойчивости для САУ угла рыскания при плоском развороте для задания на зачёт

Приложение 3

```
clc
clear

syms s k_psi_e k_omx k_sigm

% Зададим параметры в системе
H = 5;
M = 0.6;
V = 192;
Zb = -0.2;
sin = 0.08;
cos = 1;
g_del_V = 0.051;

M_x_b = -5.8;
M_x_omx = -1;
M_x_omy = -0.2;
M_x_sigm = -7;
M_y_b = -3;
M_y_omx = -0.05;
M_y_omy = -0.2;
M_y_sigmn = -2.5;

om_pr = 20;
kzi_pr = 0.707;

% Передаточная функция привода
W_pr = om_pr^2 / (s^2 + 2*om_pr*kzi_pr*s + om_pr^2);
% Передаточная функция ОУ
W1 = M_x_sigm / (s - M_x_omx);
% Передаточная функция ОУ и привода вместе
W2 = collect(W_pr*W1);
% Передаточная функция с обратной связью
W3 = collect(W2 / (1 - W2*k_omx));
% Умножаем на интегратор
W4 = collect(W3 / s);
% Ещё раз оборачиваем обратной связью
W5 = collect(W4 / (1 - W4*k_sigm));
% ПФ разомкнутой цепи
W6 = collect(k_psi_e * W5 * g_del_V / s);
% ПФ всей системы
W7 = collect(W6 / (1 - W6));

% Выделим числитель и знаменатель итоговой ПФ
[num, den] = numden(W7);
num = collect(num);
den = collect(den);
% Выделим коэффициенты числителя и знаменателя
% При этом сами коэффициенты будут записаны в порядке от младшего к
% старшему, то есть перевернутся
koefs_num = coeffs(num, 's');
koefs_den = coeffs(den, 's');
% Выделим старший коэффициент знаменателя. В списке он будет последний
starsh_koef = koefs_den(length(koefs_den));
% Поделим все коэффициенты на это число
koefs_num = vpa(koefs_num / starsh_koef);
koefs_den = vpa(koefs_den / starsh_koef);
% Развернём оба списка с коэффициентами, чтобы они шли от старшего ко
```

```

% младшему
koeffs_num = rot90(rot90(koeffs_num));
koeffs_den = rot90(rot90(koeffs_den));

% Преобразованные коэффициенты передаточной функции
B0 = koeffs_num(1);
A5 = koeffs_den(1);
A4 = koeffs_den(2);
A3 = koeffs_den(3);
A2 = koeffs_den(4);
A1 = koeffs_den(5);
A0 = koeffs_den(6);

% Передаточная функция в нормальном виде (старший коэффициент знаменателя
% равен 1
num = B0;
den = (A5*s^5 + A4*s^4 + A3*s^3 + A2*s^2 + A1*s + A0);
PF = (num / den);
disp('Числитель ПФ')
pretty(num)
disp('Знаменатель ПФ')
pretty(den)

% Зададим параметры для оптимизации в fmincon
k0 = [1;1;1];
A = [];
b = [];
Aeg = [];
beg = [];
lb = 0.1*ones(2,1);
ub = 300*ones(2,1);

% Осуществим параметрическую оптимизацию
[x, fval] = fmincon('fun_min', k0, A, b, Aeg, beg, lb, ub, 'nonclon');

% Полученные коэффициенты
k_omx = x(1)
k_sigm = x(2)
k_psi_e = x(3)
% Преобразовывает коэффициенты характеристического полинома,
% подставляя в них найденные значения
B_0 = vpa(-142.8*x(3));
A_5 = vpa(1);
A_4 = vpa(29.28);
A_3 = vpa(428.28);
A_2 = vpa(400.0 + 2800.0*x(1));
A_1 = vpa(2800.0*x(2));
A_0 = vpa(142.8*x(3));
num = B_0;
den = A_5*s^5 + A_4*s^4 + A_3*s^3 + A_2*s^2 + A_1*s + A_0;
num = sym2poly(num);
den = sym2poly(den);
% Получаем передаточную функцию и строим переходный процесс
disp('Передаточная функция с полученными коэффициентами')
W_sys = tf(num, den)
figure;
step(-1*W_sys)
grid on

```

Программа 9. Основной код для системы управления углом рыскания в режиме координированного разворота

```

function f = fun_min(x)
% x(1) = k_omegay, x(2) = k_psi

    A_1 = 2800*x(2);
    A_0 = 142.8*x(3);

    f = A_1 * A_0^(-1);

end

```

Программа 10. Целевая функция для системы управления углом рыскания в режиме
координированного разворота

```

function [c, seq] = nonclon(x)
% x(1) = k_omegay, x(2) = k_psi

    A_5 = 1;
    A_4 = 29.28;
    A_3 = 428.28;
    A_2 = 400.0 + 2800.0*x(1);
    A_1 = 2800*x(2);
    A_0 = 142.8*x(3);

    lambd_star = 2.15; % Желаемое значение лямбды
    lambd1 = lambd_star - (A_0^(-1) * A_1 * A_2 * A_3^(-1));
    lambd2 = lambd_star - (A_1^(-1) * A_2 * A_3 * A_4^(-1));
    lambd3 = lambd_star - (A_2^(-1) * A_3 * A_4 * A_5^(-1));

    sigm_star = 2.1; % Желаемое значение сигмы
    sigm1 = sigm_star - (A_0^(-1) * A_1^2 * A_2^(-1));
    sigm2 = sigm_star - (A_1^(-1) * A_2^2 * A_3^(-1));
    sigm3 = sigm_star - (A_2^(-1) * A_3^2 * A_4^(-1));

    c = [lambd1; lambd2; lambd3; sigm1; sigm2; sigm3];
    seq = [];

end

```

Программа 11. Ограничения для системы управления углом рыскания в режиме
координированного разворота

```

clc
clear

syms k_psi_e k_omx k_sigm s

% Зададим параметры в системе
H = 5;
M = 0.6;
V = 192;
Zb = 0.2;
sin = 0.08;
cos = 1;
g_del_V = 0.051;

M_x_b = -5.8;
M_x_omx = -1;
M_x_omy = -0.2;

```

```

M_x_signe = -7;
M_y_b = -3;
M_y_omx = -0.05;
M_y_omy = -0.2;
M_y_sigmn = -2.5;

om_pr = 20;
kzi_pr = 0.707;

% Передаточная функция привода
W_pr = om_pr^2 / (s^2 + 2*om_pr*kzi_pr*s + om_pr^2);
% Передаточная функция ОУ
W1 = M_x_signe / (s - M_x_omx);
% Передаточная функция ОУ и привода вместе
W2 = collect(W_pr*W1);
% Передаточная функция с обратной связью
W3 = collect(W2 / (1 - W2*k_omx));
% Умножаем на интегратор
W4 = collect(W3 / s);
% Ещё раз оборачиваем обратной связью
W5 = collect(W4 / (1 - W4*k_sigmn));
% ПФ разомкнутой цепи
W6 = collect(k_psi_e * W5 * g_del_V / s);
% ПФ всей системы
W7 = collect(W6 / (1 - W6));

% Выделим числитель и знаменатель итоговой ПФ
[num, den] = numden(W7);
num = collect(num);
den = collect(den);
% Выделим коэффициенты числителя и знаменателя
% При этом сами коэффициенты будут записаны в порядке от младшего к
% старшему, то есть перевернутся
koefs_num = coeffs(num, 's');
koefs_den = coeffs(den, 's');
% Выделим старший коэффициент знаменателя. В списке он будет последний
starsh_koef = koefs_den(length(koefs_den));
% Поделим все коэффициенты на это число
koefs_num = vpa(koefs_num / starsh_koef);
koefs_den = vpa(koefs_den / starsh_koef);
% Развернём оба списка с коэффициентами, чтобы они шли от старшего ко
% младшему
koefs_num = rot90(rot90(koefs_num));
koefs_den = rot90(rot90(koefs_den));

% Преобразованные коэффициенты передаточной функции
B0 = koefs_num(1);
A5 = koefs_den(1);
A4 = koefs_den(2);
A3 = koefs_den(3);
A2 = koefs_den(4);
A1 = koefs_den(5);
A0 = koefs_den(6);

% Передаточная функция в нормальном виде (старший коэффициент знаменателя
% равен 1
num = B0;
den = (A5*s^5 + A4*s^4 + A3*s^3 + A2*s^2 + A1*s + A0);
PF = (num / den);
disp('Числитель ПФ')
pretty(num)
disp('Знаменатель ПФ')

```

```

pretty(den)

% Запишем неравенства для критерия Рауса
eq1 = A0 > 0;
eq2 = A1 > 0;
eq3 = A2 > 0;
eq4 = A3 > 0;
eq5 = A4 > 0;
eq6 = A5 > 0;
eq7 = simplify((A1*A2-A0*A3)*(A3*A4-A2*A5) - (A1*A4-A0*A5)^2) > 0;

% Преобразуем неравенства из символьного типа данных в функции
eq1 = matlabFunction(eq1);
eq2 = matlabFunction(eq2);
eq3 = matlabFunction(eq3);
eq4 = matlabFunction(eq4);
eq5 = matlabFunction(eq5);
eq6 = matlabFunction(eq6);
eq7 = matlabFunction(eq7);

% Задание сетки
ko = [-1:0.05:4];
ks = [-1:0.05:20];
kp = [-1:0.5:300];

% Построение трехмерного графика
% Задаём k_omega
% Строим плоскую область для параметров k_sigm, k_psi
% Прибавляет к параметру k_omega шаг 0.05
% Повторяем действия
% Таким образом - получится много плоских рисунков,
% которые в сумме будут давать пространство устойчивости
hold on
grid on
view(3)
for i=1:length(ko)
    X = [];
    Y = [];
    Z = [];
    ind = 1;
    for j=1:length(ks)
        for k=1:length(kp)
            c1 = eq1(kp(k));
            c2 = eq2(ks(j));
            c3 = eq3(ko(i));
            c4 = eq4();
            c5 = eq5();
            c6 = eq6();
            c7 = eq7(ko(i), kp(k), ks(j));
            if c1 & c2 & c3 & c4 & c5 & c6 & c7
                X(ind) = ko(i);
                Y(ind) = ks(j);
                Z(ind) = kp(k);
                ind = ind + 1;
            end
        end
    end
    plot3(X, Y, Z)
end

```



```

xlabel('Komx')
ylabel('Ksigm')
zlabel('Kpsie')

```

Программа 12. Построение области устойчивости для системы управления углом рыскания в режиме координированного разворота

```

clc
clear

syms k_psi_e k_omx k_sigm s

% Зададим параметры в системе
H = 5;
M = 0.6;
V = 192;
Zb = 0.2;
sin = 0.08;
cos = 1;
g_del_V = 0.051;

M_x_b = -5.8;
M_x_omx = -1;
M_x_omy = -0.2;
M_x_sigme = -7;
M_y_b = -3;
M_y_omx = -0.05;
M_y_omy = -0.2;
M_y_sigmn = -2.5;

om_pr = 20;
kzi_pr = 0.707;

% Передаточная функция привода
W_pr = om_pr^2 / (s^2 + 2*om_pr*kzi_pr*s + om_pr^2);
% Передаточная функция ОУ
W1 = M_x_sigme / (s - M_x_omx);
% Передаточная функция ОУ и привода вместе
W2 = collect(W_pr*W1);
% Передаточная функция с обратной связью
W3 = collect(W2 / (1 - W2*k_omx));
% Умножаем на интегратор
W4 = collect(W3 / s);
% Ещё раз оборачиваем обратной связью
W5 = collect(W4 / (1 - W4*k_sigm));
% ПФ разомкнутой цепи
W6 = collect(k_psi_e * W5 * g_del_V / s);
% ПФ всей системы
W7 = collect(W6 / (1 - W6));

% Выделим числитель и знаменатель итоговой ПФ
[num, den] = numden(W7);
num = collect(num);
den = collect(den);
% Выделим коэффициенты числителя и знаменателя
% При этом сами коэффициенты будут записаны в порядке от младшего к
% старшему, то есть перевернутся
koefs_num = coeffs(num, 's');
koefs_den = coeffs(den, 's');

```

```

% Выделим старший коэффициент знаменателя. В списке он будет последний
starsh_koef = koefs_den(length(koefs_den));
% Поделим все коэффициенты на это число
koefs_num = vpa(koefs_num / starsh_koef);
koefs_den = vpa(koefs_den / starsh_koef);
% Развернём оба списка с коэффициентами, чтобы они шли от старшего ко
% младшему
koefs_num = rot90(rot90(koefs_num));
koefs_den = rot90(rot90(koefs_den));

% Преобразованные коэффициенты передаточной функции
B0 = koefs_num(1);
A5 = koefs_den(1);
A4 = koefs_den(2);
A3 = koefs_den(3);
A2 = koefs_den(4);
A1 = koefs_den(5);
A0 = koefs_den(6);

% Передаточная функция в нормальном виде (старший коэффициент знаменателя
% равен 1
num = B0;
den = (A5*s^5 + A4*s^4 + A3*s^3 + A2*s^2 + A1*s + A0);
PF = (num / den);
disp('Числитель ПФ')
pretty(num)
disp('Знаменатель ПФ')
pretty(den)

% Запишем неравенства для достаточного условия устойчивости
lambd_star = 2.15; % Желаемое значение лямбды
eq1d = A0^(-1) * A1 * A2 * A3^(-1) >= lambd_star;
eq2d = A1^(-1) * A2 * A3 * A4^(-1) >= lambd_star;
eq3d = A2^(-1) * A3 * A4 * A5^(-1) >= lambd_star;

% Преобразуем неравенства из символьного типа данных в функции
eq1d = matlabFunction(eq1d);
eq2d = matlabFunction(eq2d);
eq3d = matlabFunction(eq3d);

% Задание сетки
ko = [-1:0.05:8];
ks = [-1:0.05:20];
kp = [0.1:0.5:300];

disp('Строим график по достаточным условиям устойчивости')
% Построение трехмерного графика
% Задаём k_omega
% Строим плоскую область для параметров k_omega, k_sigma
% Прибавляет к параметру k_psi шаг
% Повторяем действия
% Таким образом - получится много плоских рисунков,
% которые в сумме будут давать пространство устойчивости
figure;
hold on
grid on
view(3)
colors = {'red', 'yellow'};
tic % Начинаем считать время
for i=1:length(kp)
    X = [];

```

```

Y = [];
Z = [];
ind = 1;
for j=1:length(ko)
    for k=1:length(ks)
        c1 = eq1d(ko(j), kp(i), ks(k));
        c2 = eq2d(ko(j), ks(k));
        c3 = eq3d(ko(j));
        if c1 & c2 & c3
            X(ind) = ko(j);
            Y(ind) = ks(k);
            Z(ind) = kp(i);
            ind = ind + 1;
        end
    end
end
col = colors(randi([1, length(colors)]));
col = string(col);
plot3(X, Y, Z, Color=col)
end
toc % Заканчиваем считать время

% Запишем неравенства ждя критерия Рауса
eq1r = A0 > 0;
eq2r = A1 > 0;
eq3r = A2 > 0;
eq4r = A3 > 0;
eq5r = A4 > 0;
eq6r = A5 > 0;
eq7r = simplify((A1*A2-A0*A3)*(A3*A4-A2*A5) - (A1*A4-A0*A5)^2) > 0;

% Преобразуем неравенства из символьного типа данных в функции
eq1r = matlabFunction(eq1r);
eq2r = matlabFunction(eq2r);
eq3r = matlabFunction(eq3r);
eq4r = matlabFunction(eq4r);
eq5r = matlabFunction(eq5r);
eq6r = matlabFunction(eq6r);
eq7r = matlabFunction(eq7r);

% Задание сетки
ko = [-1:0.1:4];
ks = [-1:0.1:20];
kp = [0.1:30:300];

% Построение трехмерного графика
% Задаём k_omega
% Строим плоскую область для параметров k_sigm, k_psi
% Прибавляет к параметру k_omega шаг 0.05
% Повторяем действия
% Таким образом - получится много плоских рисунков,
% которые в сумме будут давать пространство устойчивости
hold on
grid on
view(3)
colors = {'green', 'blue'};
disp('Строим график по критериям Рауса')
tic % Начинаем считать время
for i=1:length(kp)
    X = [];
    Y = [];

```

```

Z = [];
ind = 1;
for j=1:length(ko)
    for k=1:length(ks)
        c1 = eq1r(kp(i));
        c2 = eq2r(ks(k));
        c3 = eq3r(ko(j));
        c4 = eq4r();
        c5 = eq5r();
        c6 = eq6r();
        c7 = eq7r(ko(j), kp(i), ks(k));
        if c1 & c2 & c3 & c4 & c5 & c6 & c7
            X(ind) = ko(j);
            Y(ind) = ks(k);
            Z(ind) = kp(i);
            ind = ind + 1;
        end
    end
end
col = colors(randi([1, length(colors)]));
col = string(col);
plot3(X, Y, Z, Color=col)
end
toc % Заанчиваем считать время
xlabel('Komx')
ylabel('Ksigm')
zlabel('Kpsie')
hold off

clc
clear
syms k_psi_e k_omx k_sigm s

% Зададим параметры в системе
H = 5;
M = 0.6;
V = 192;
Zb = 0.2;
sin = 0.08;
cos = 1;
g_del_V = 0.051;

M_x_b = -5.8;
M_x_omx = -1;
M_x_omy = -0.2;
M_x_sigme = -7;
M_y_b = -3;
M_y_omx = -0.05;
M_y_omy = -0.2;
M_y_sigmn = -2.5;

om_pr = 20;
kzi_pr = 0.707;

% Передаточная функция привода
W_pr = om_pr^2 / (s^2 + 2*om_pr*kzi_pr*s + om_pr^2);
% Передаточная функция ОУ
W1 = M_x_sigme / (s - M_x_omx);
% Передаточная функция ОУ и привода вместе
W2 = collect(W_pr*W1);
% Передаточная функция с обратной связью
W3 = collect(W2 / (1 - W2*k_omx));

```

```

% Умножаем на интегратор
W4 = collect(W3 / s);
% Ещё раз оборачиваем обратной связью
W5 = collect(W4 / (1 - W4*k_sigm));
% ПФ разомкнутой цепи
W6 = collect(k_psi_e * W5 * g_del_V / s);
% ПФ всей системы
W7 = collect(W6 / (1 - W6));

% Выделим числитель и знаменатель итоговой ПФ
[num, den] = numden(W7);
num = collect(num);
den = collect(den);
% Выделим коэффициенты числителя и знаменателя
% При этом сами коэффициенты будут записаны в порядке от младшего к
% старшему, то есть перевернутся
koefs_num = coeffs(num, 's');
koefs_den = coeffs(den, 's');
% Выделим старший коэффициент знаменателя. В списке он будет последний
starsh_koef = koefs_den(length(koefs_den));
% Поделим все коэффициенты на это число
koefs_num = vpa(koefs_num / starsh_koef);
koefs_den = vpa(koefs_den / starsh_koef);
% Развернём оба списка с коэффициентами, чтобы они шли от старшего ко
% младшему
koefs_num = rot90(rot90(koefs_num));
koefs_den = rot90(rot90(koefs_den));

% Преобразованные коэффициенты передаточной функции
B0 = koefs_num(1);
A5 = koefs_den(1);
A4 = koefs_den(2);
A3 = koefs_den(3);
A2 = koefs_den(4);
A1 = koefs_den(5);
A0 = koefs_den(6);

% Передаточная функция в нормальном виде (старший коэффициент знаменателя
% равен 1
num = B0;
den = (A5*s^5 + A4*s^4 + A3*s^3 + A2*s^2 + A1*s + A0);
PF = (num / den);
disp('Числитель ПФ')
pretty(num)
disp('Знаменатель ПФ')
pretty(den)

% Запишем неравенства для достаточного условия устойчивости
lambd_star = 2.15; % Желаемое значение лямбды
eq1d = A0^(-1) * A1 * A2 * A3^(-1) >= lambd_star;
eq2d = A1^(-1) * A2 * A3 * A4^(-1) >= lambd_star;
eq3d = A2^(-1) * A3 * A4 * A5^(-1) >= lambd_star;

% Преобразуем неравенства из символьного типа данных в функции
eq1d = matlabFunction(eq1d);
eq2d = matlabFunction(eq2d);
eq3d = matlabFunction(eq3d);

% Задание сетки
ko = [-1:0.01:4];
ks = [-1:0.01:15];

```

```

kp = [30:50:300];
[K0, KS] = meshgrid(ko, ks);
figure;
hold on;
grid on;
colors = {'green', 'blue', 'yellow', 'blue', 'magenta', 'red', 'black'};
ind_col = 1;
for i=1:length(kp)
    eq1 = eq1d(k_omx, kp(i), k_sigm);
    eq2 = eq2d(k_omx, k_sigm);
    eq3 = eq3d(k_omx);
    eq1 = matlabFunction(eq1);
    eq2 = matlabFunction(eq2);
    eq3 = matlabFunction(eq3);
    c1 = eq1(K0, KS);
    c2 = eq2(K0, KS);
    c3 = eq3(K0);
    col = colors(ind_col);
    col = string(col);
    ind_col = ind_col + 1;
    contourf(ko, ks, c1 & c2 & c3, [1 1 1], FaceAlpha=0.2, FaceColor=col)
    colormap lines
end
hold off
xlabel('Komx')
ylabel('Ksigm')
title('Область устойчивости по достаточным условиям для kpsie=30, 80, 130, 180, 230, 280')

% Запишем неравенства для достаточного условия устойчивости
eq1r = A0 > 0;
eq2r = A1 > 0;
eq3r = A2 > 0;
eq4r = A3 > 0;
eq5r = A4 > 0;
eq6r = A5 > 0;
eq7r = simplify((A1*A2-A0*A3)*(A3*A4-A2*A5) - (A1*A4-A0*A5)^2) > 0;

% Преобразуем неравенства из символьного типа данных в функции
eq1r = matlabFunction(eq1r);
eq2r = matlabFunction(eq2r);
eq3r = matlabFunction(eq3r);
eq4r = matlabFunction(eq4r);
eq5r = matlabFunction(eq5r);
eq6r = matlabFunction(eq6r);
eq7r = matlabFunction(eq7r);

% Задание сетки
ko = [-1:0.01:4];
ks = [-1:0.01:18];
kp = [30:50:300];
[K0, KS] = meshgrid(ko, ks);
figure;
hold on;
grid on;
colors = {'green', 'blue', 'yellow', 'magenta', 'red', 'black'};
ind_col = 1;
for i=1:length(kp)
    eq2 = eq2r(k_sigm);
    eq3 = eq3r(k_omx);
    eq7 = eq7r(k_omx, kp(i), k_sigm);

```

```

eq2 = matlabFunction(eq2);
eq3 = matlabFunction(eq3);
eq7 = matlabFunction(eq7);
c1 = eq2(KS);
c2 = eq3(K0);
c3 = eq7(K0, KS);
col = colors(ind_col);
col = string(col);
ind_col = ind_col + 1;
contourf(ko, ks, c1 & c2 & c3, [1 1 1], FaceAlpha=0.2, FaceColor=col)
end
hold off
xlabel('Комх')
ylabel('Ksigm')
title('Область устойчивости по критерию Рауса для kpsie=30, 80, 130, 180, 230, 280')

% Рабочая точка
komx = 0.9225;
ksigm = 3.5337;
kpsie = 109.4343;

figure;
grid on;
hold on;
ko = [-1:0.01:4];
ks = [-1:0.01:15];
[K0, KS] = meshgrid(ko, ks);
eq1 = eq1d(k_omx, kpsie, k_sigm);
eq2 = eq2d(k_omx, k_sigm);
eq3 = eq3d(k_omx);
eq1 = matlabFunction(eq1);
eq2 = matlabFunction(eq2);
eq3 = matlabFunction(eq3);
c1 = eq1(K0, KS);
c2 = eq2(K0, KS);
c3 = eq3(K0);
contourf(ko, ks, c1 & c2 & c3, [1 1 1], FaceAlpha=0.2, FaceColor='red')

ko = [-1:0.01:4];
ks = [-1:0.01:18];
kp = [30:50:300];
[K0, KS] = meshgrid(ko, ks);
eq2 = eq2r(k_sigm);
eq3 = eq3r(k_omx);
eq7 = eq7r(k_omx, kpsie, k_sigm);
eq2 = matlabFunction(eq2);
eq3 = matlabFunction(eq3);
eq7 = matlabFunction(eq7);
c1 = eq2(KS);
c2 = eq3(K0);
c3 = eq7(K0, KS);
contourf(ko, ks, c1 & c2 & c3, [1 1 1], FaceAlpha=0.2, FaceColor='green')

scatter(komx, ksigm, 'filled')
hold off
xlabel('Комх')
ylabel('Ksigm')
title('Область устойчивости по двум критериям для kpsie=109.4343')

```

Программа 13. Программа для областей устойчивости для САУ угла рыскания при координированном развороте для задания на зачёт