



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

Факультет: «Специальное машиностроение»

Кафедра: «Робототехнические системы и мехатроника»

## **Лабораторная работа № 3**

по курсу «Теория автоматического управления»

Вариант 14

Выполнил: Садовец Роман  
Группа: СМ7-51Б

Проверил(а):

Москва, 2023 г.

# I. Модель двигателя постоянного тока

а) Двигатель постоянного тока (далее ДПТ) описывается следующей системой уравнений:

$$\begin{cases} J_{\text{дв}} \ddot{\phi} = M_{\text{дв}} - M_{\text{тр}} - M_{\text{нагр}}; \\ M_{\text{дв}} = k_M I_{\text{я}}; \\ L_{\text{я}} \dot{I}_{\text{я}} + R_{\text{я}} I_{\text{я}} = U_{\text{упр}} - E_{\text{противо}}; \\ E_{\text{противо}} = k_{\omega} \dot{\phi} \end{cases} \quad (1)$$

Для простоты примем, что двигатель работает в режиме холостого хода ( $M_{\text{нагр}} = 0$ ), а момент трения пренебрежимо мал ( $M_{\text{тр}} = 0$ ). Тогда система примет следующий вид:

$$\begin{cases} J_{\text{дв}} \ddot{\phi} = M_{\text{дв}}; \\ M_{\text{дв}} = k_M I_{\text{я}}; \\ L_{\text{я}} \dot{I}_{\text{я}} + R_{\text{я}} I_{\text{я}} = U_{\text{упр}} - E_{\text{противо}}; \\ E_{\text{противо}} = k_{\omega} \dot{\phi} \end{cases} \quad (2)$$

Для заданного варианта параметры двигателя представлены на рисунке 1.

Тип двигателя	P <sub>д ном</sub> Вт	N <sub>д ном</sub> об/мин	U <sub>я ном</sub> В	I <sub>я ном</sub> А	R <sub>я</sub> Ом	J <sub>д</sub> *10 <sup>-6</sup> кг*м <sup>2</sup>	m кг	L <sub>кд</sub>
МИГ-90Б	90	6000	27	4,1	0,7	7,9	2,0	9,4

Рис. 1. Параметры двигателя

Рассчитаем параметры МИГ-90Б - ДПТ с независимым возбуждением:

$$\omega_{\text{ном}} = \pi * n_{\text{ном}} / 30 = 3,14 * 6000 / 30 \approx 628 \text{ рад}^{-1};$$

$$M_{\text{ном}} = P_{\text{ном}} / \omega_{\text{ном}} = 90 / 628 \approx 0,143 \text{ Н * м};$$

$$k_M = M_{\text{ном}} / I_{\text{я ном}} = 0,143 / 4,1 \approx 0,035; \quad (3)$$

$$k_{\omega} = (U_{\text{я ном}} - I_{\text{я ном}} * R_{\text{я}}) / \omega_{\text{ном}} = (27 - 4,1 * 0,7) / 628 \approx 0,038; \quad (4)$$

$$\begin{aligned} L_{\text{я}} &\approx 0,1 * (J_{\text{дв}} * R_{\text{я}}^2) / (k_{\omega} * k_M) = 0,1 * (7,9 * 10^{-6} * 0,7^2) / (0,038 * 0,035) \approx \\ &\approx 2,91 * 10^{-4}; \end{aligned} \quad (5)$$

По условию обороты ДПТ управляются с помощью входного напряжения  $U$ . Исходя из третьего выражения системы (2), в системе присутствуют обратная связь  $E_{\text{противо}}$ , что показано сумматором на рисунке 2. Далее выражаем  $I_{\text{я}}$  через третье уравнение системы с помощью передаточной функции:

$$W_1(s) = I_{\text{я}} / (U_{\text{упр}} - E_{\text{противо}}) = 1 / (L_{\text{я}} * s + R_{\text{я}}) = k_1 / (T_1 * s + 1); \quad (6)$$

где  $T_1 = L_{\text{я}} / R_{\text{я}} = 2,91 * 10^{-4} / 0,7 \approx 4,16 * 10^{-4}$ ;  $k_1 = 1 / R_{\text{я}} = 1,43$

Для получения момента двигателя необходимо  $I_{\text{я}}$  домножить на  $k_{\text{М}}$  (показано усилительным звеном). С учетом полученного момента двигателя  $M_{\text{дв}}$  можно из первого уравнения системы выразить ускорение вала двигателя с помощью следующей передаточной функции:

$$W_2(s) = \ddot{\varphi} / M_{\text{дв}} = 1 / J_{\text{дв}} \quad (7)$$

На выходе получаем угловое ускорение, интегрируя можно найти угловую скорость, второй раз - искомый угол  $\varphi$ . Однако не стоит забывать и про ООС. Для её получения воспользуемся 4 выражением в системе (2): домножим  $\dot{\varphi}$  на значение  $k_{\omega}$  и получим  $E_{\text{противо}}$

Перенесем полученную систему в Simulink (рис. 2)

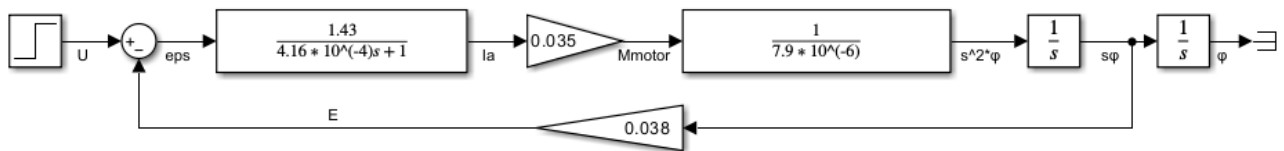


Рис. 2. Структурная схема ДПТ в среде разработки Simulink

Подадим на вход единичное напряжение в течение 0.03 секунд. Полученный график представлен на рисунке 3.

Далее необходимо получить графики углового положения вала, угловой скорости, ускорения и сигнала ошибки различными способом. Начнем с блока Scope. Переделенная схема представлена на рисунке 4, а на рисунке 5 - полученные графики.

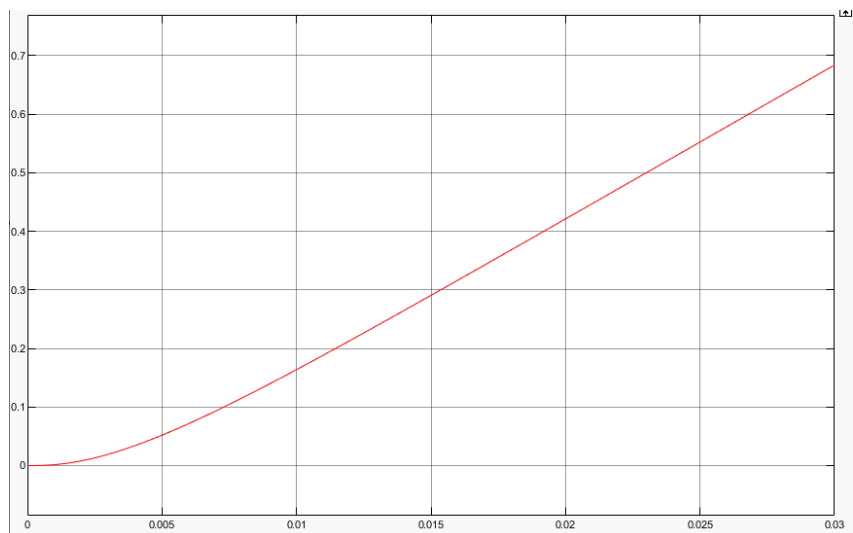


Рис. 3. График  $\varphi(t)$

Для второго способа (логирования сигналов), необходимо: 1 – выделить интересующую нас схему (дополнительно ввел выходы сигналов для облегчения их отслеживания); 2 – нажать «логирование сигналов»; 3 – запустить симуляцию; 4 – зайти в Data Inspector и включить необходимые графики для демонстрации. На рисунке 6 показана схема действия и структурная схема, а на рисунке 7 графики (кроме ускорения)

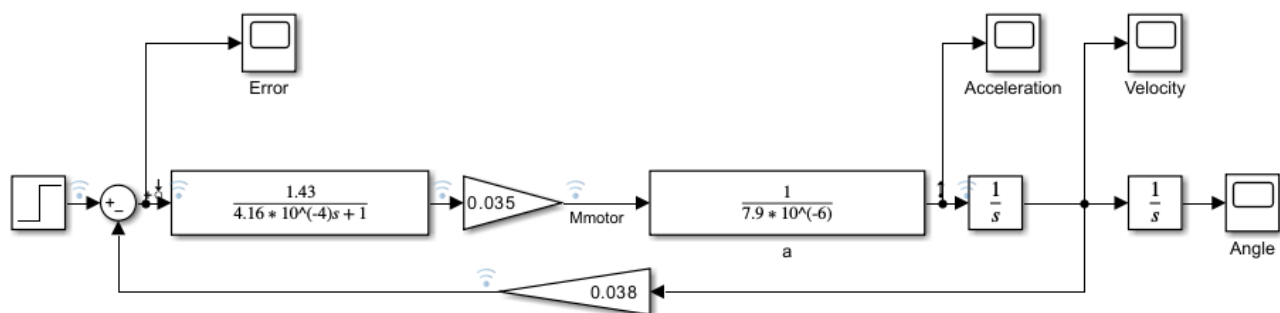
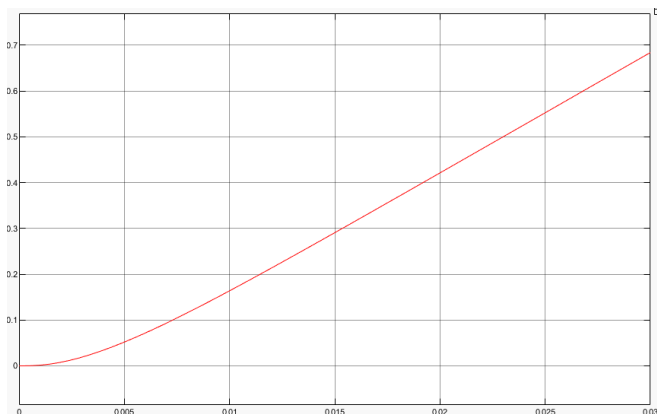
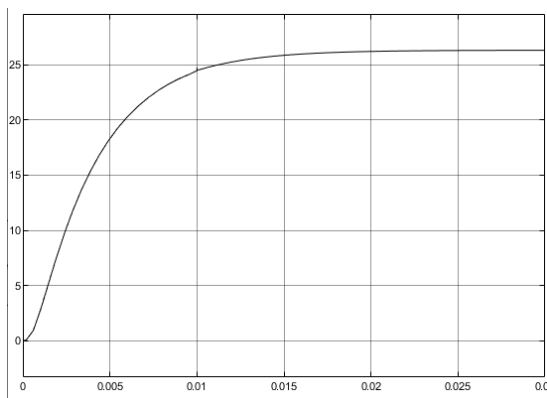


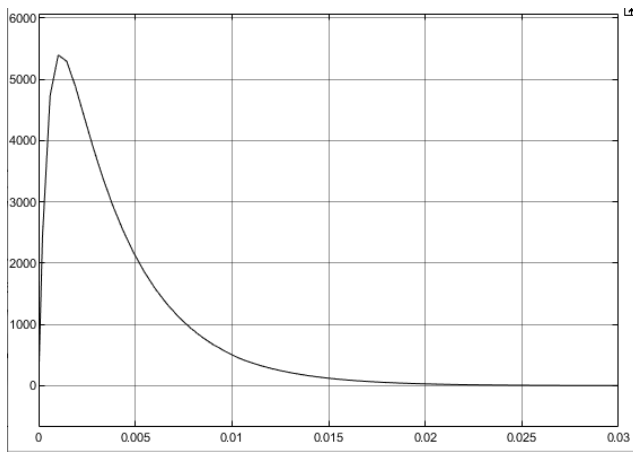
Рис. 4. Структурная схема для вывода необходимых параметров



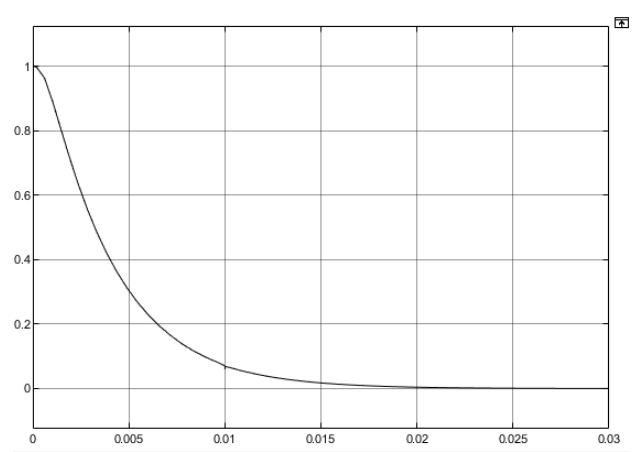
а)



б)



в)



г)

Рис. 5. Графики: а)  $\varphi(t)$ ; б)  $\dot{\varphi}(t)$ ; в)  $\ddot{\varphi}(t)$ ; г) Ошибки

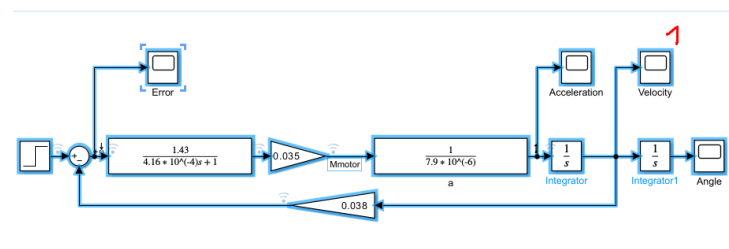
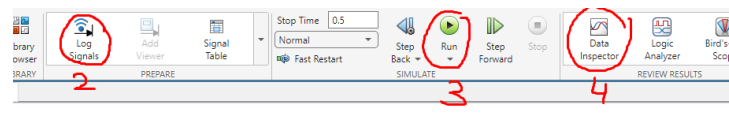


Рис. 6. Принцип действий при логировании сигналов

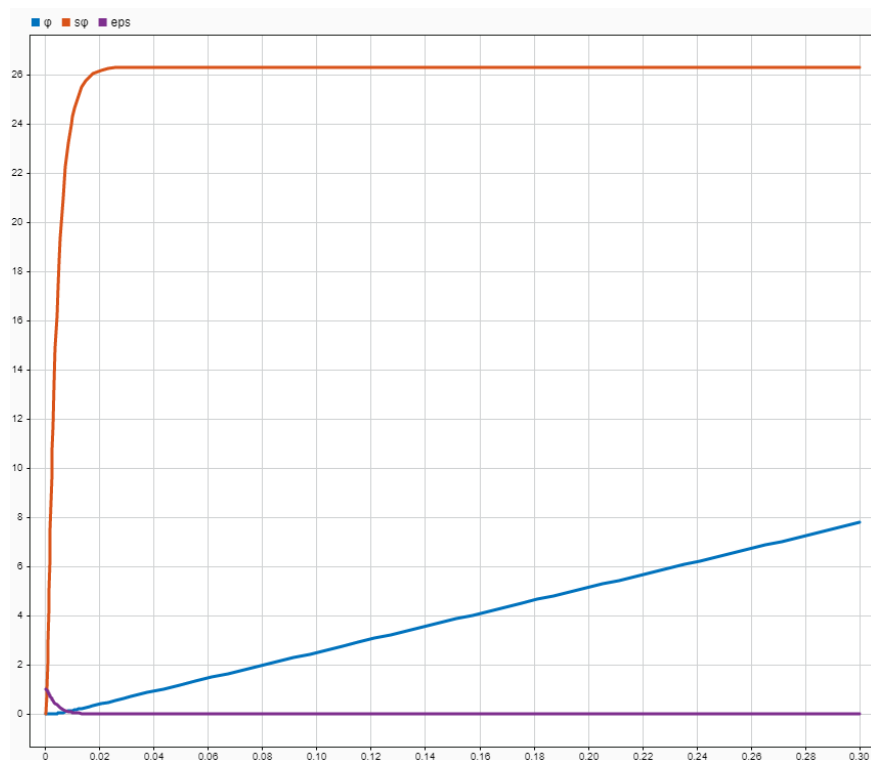


Рис. 7. Графики при логировании сигналов (кроме ускорения)

Последним способом для отображения выходных графиков является использование инструмента Model Linearizer. Он находится в разделе Apps (рис. 8). Для этого расставляем через ПКМ выходы и входы системы, которые мы будем отображать в графиках (зависимые и независимые переменные) (рис. 9). Input measurement – входной сигнал (независимая переменная), при том учитывающая предыдущие до этого сигналы; Output measurement – для выходного сигнала. Далее подаем входное единичное воздействие, нажимая в Model Linearizer

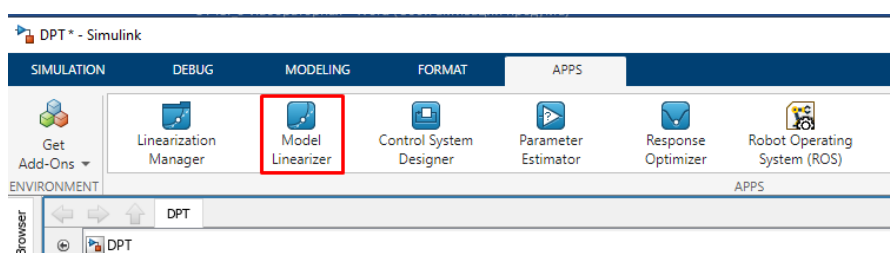


Рис. 8. Model Linearizer

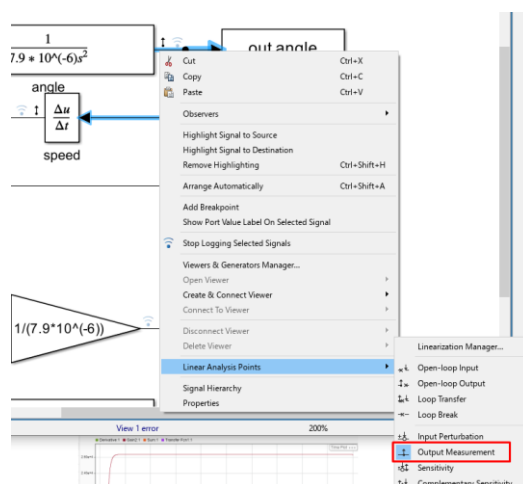


Рис. 9. Выбор входного/выходного сигнала

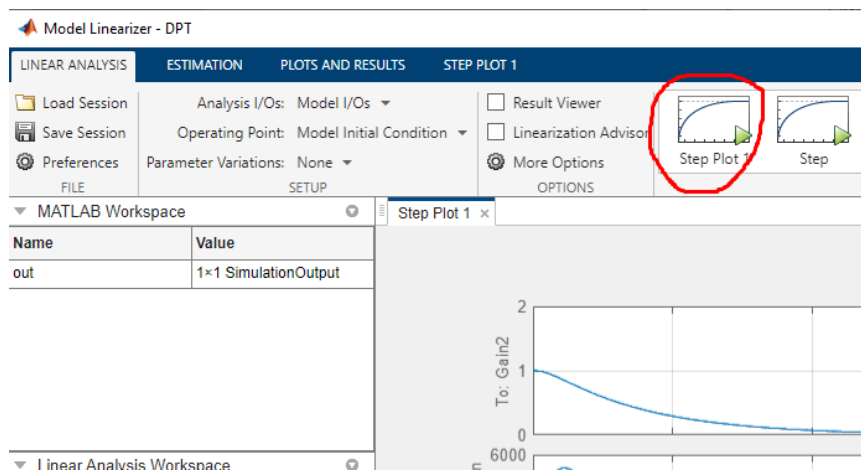


Рис. 10. Отрисовка графика единичного воздействия

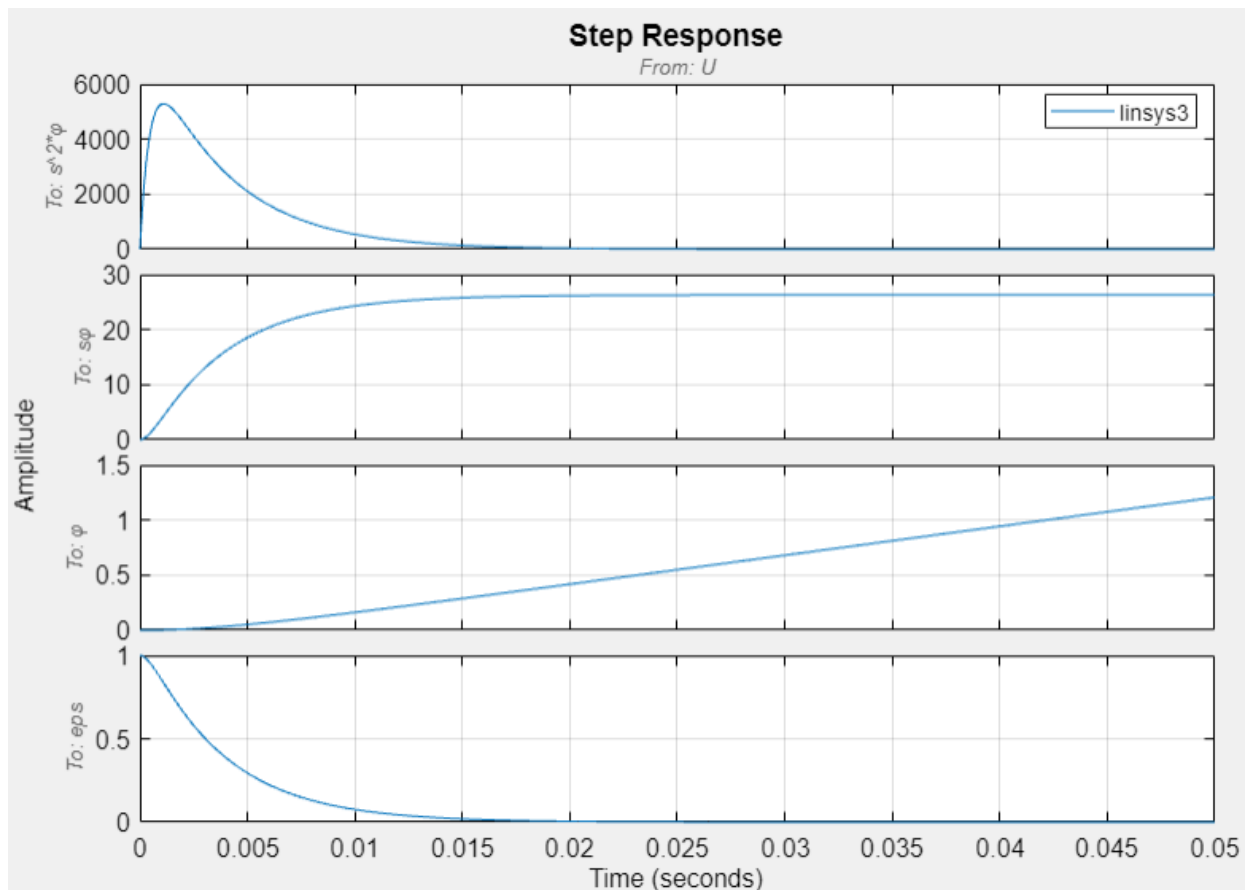


Рис. 11. Полученные графики после вызова Model Linearizer-a

Следующим шагом необходимо подать линейно изменяющийся сигнал. Для этого в структурной схеме заменяем единичный входной сигнал на Ramp с заданными параметрами (рис. 11). Параметр Slope отвечает за наклон входного сигнала. Тогда на выходе получим графики, показанные на рисунке 12.

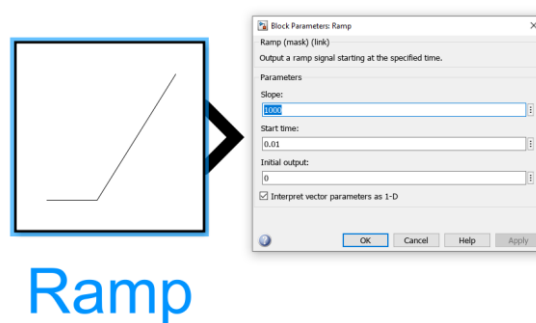


Рис. 12. Линейный входной сигнал. Настройки блока Ramp

Оставшийся сигнал – квадратичный. Реализуется с помощью блока Ramp (подача линейно изменяющегося сигнала) и Math Function – возведение в квадрат (рис. 14). Подаем этот сигнал на вход и получаем графики, показанные на рисунке 15.

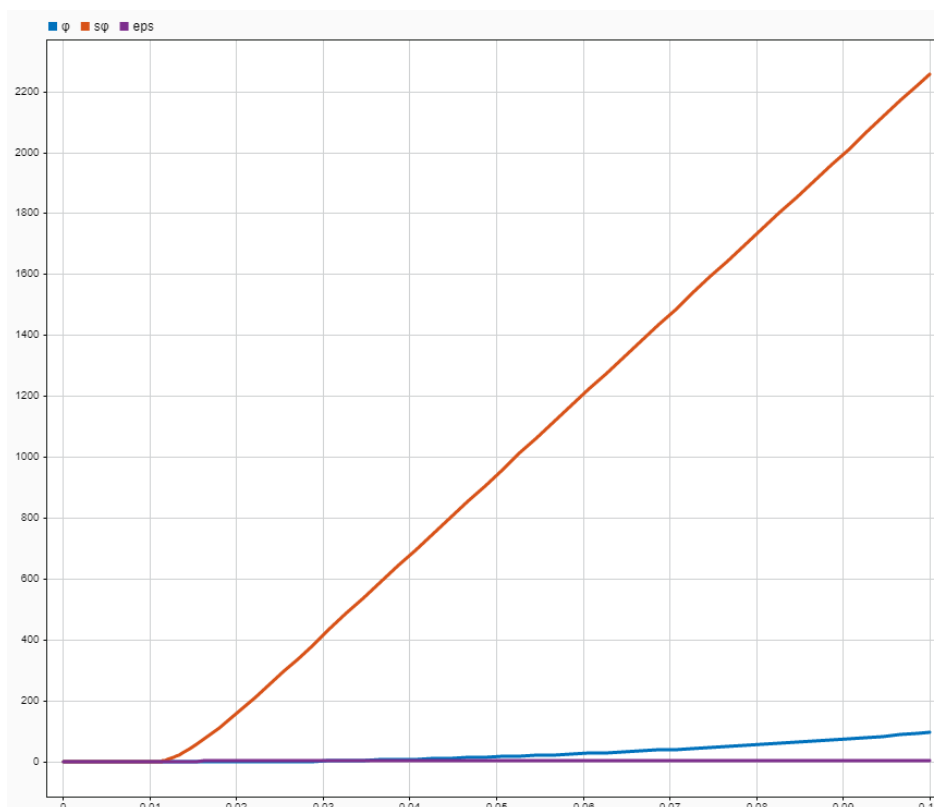


Рис. 13. График при линейном входном сигнале

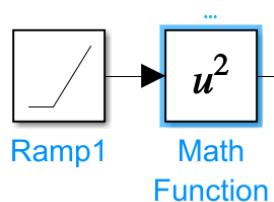


Рис. 14. Квадратичный входной сигнал. Настройка блоков

По определению Астатическая система регулирования – это такая система автоматического регулирования режимов работы промышленных установок, систем автоматического управления и др., в которой ошибка регулирования стремится к нулю независимо от размера воздействия, если последнее принимает установившееся постоянное значение

**Вывод:** Исходя из полученных графиков (рис. 16) можно смело утверждать, что исходная система управления ДПТ имеет **первый порядок астатизма**



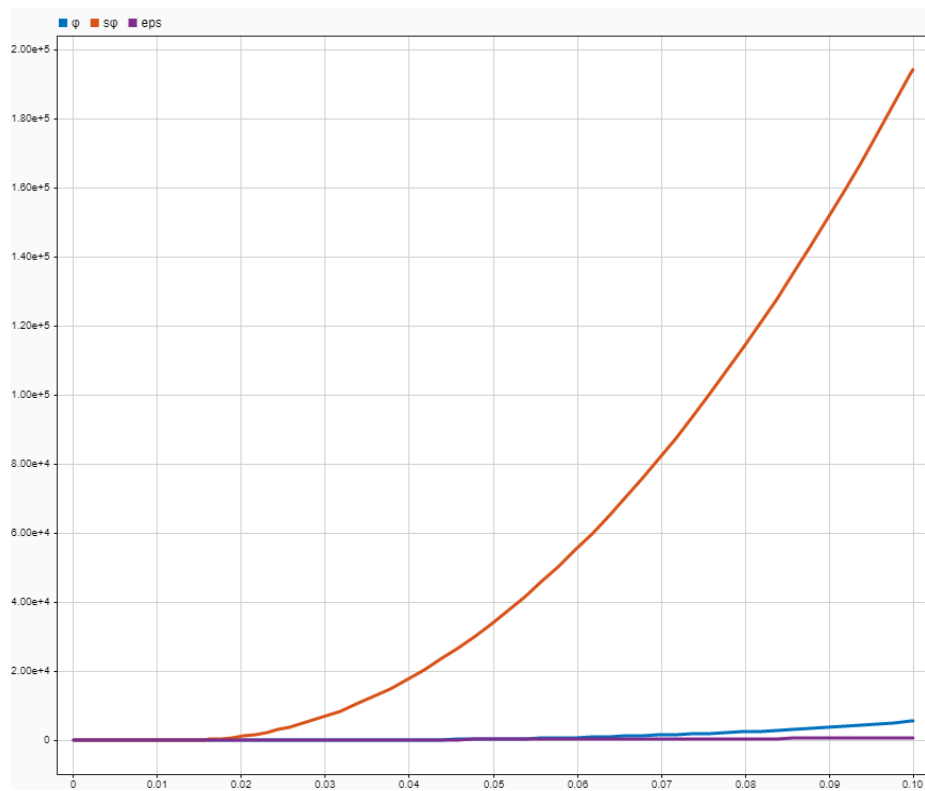
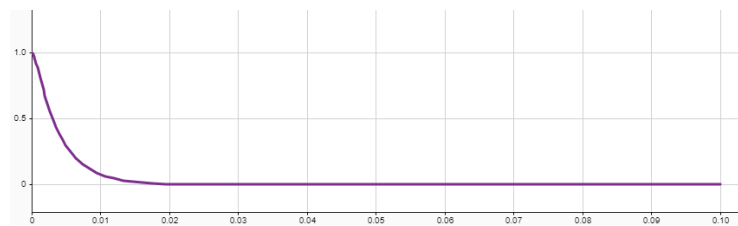
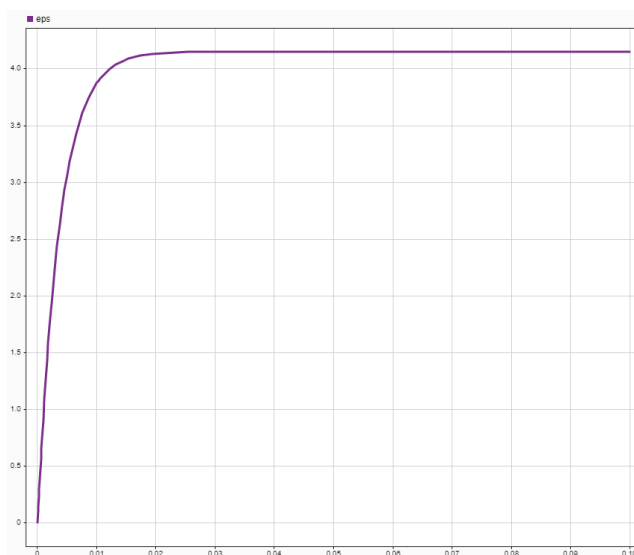


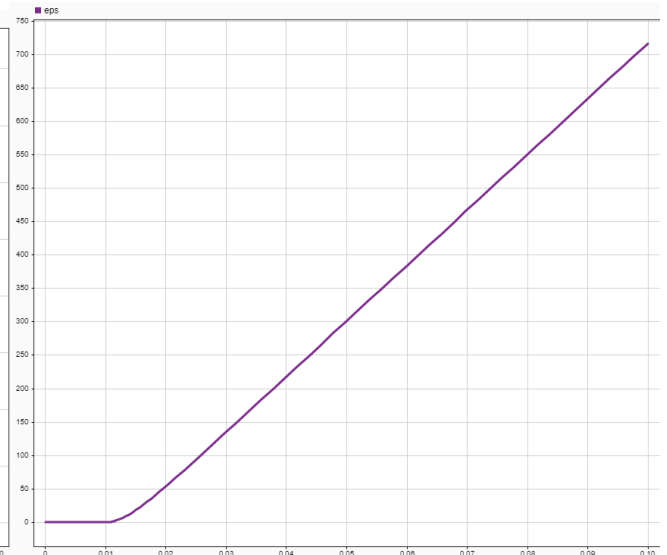
Рис. 15. График при квадратичном входном сигнале (линии аналогичны рис. 12)



а)



б)



в)

Рис. 16. Графики ошибки от: а – единичного входного сигнала, б – линейного, в - квадратичного

б) В первом пункте мы упростили систему тем, что не учитывали момент трения  $M_{тр}$  и момент нагрузки на вал  $M_{нагр}$ . Возьмем их в учет. Тогда:

$$M_{тр} = k_{тр1} * \dot{\varphi} \quad (8)$$

$$M_{нагр} = J_{нагр} \ddot{\varphi} + i * k_{тр2} * \dot{\varphi} \quad (9)$$

где  $k_{тр1}$  – коэффициент вязкого трения вала двигателя о подшипники,  $J_{нагр}$  – момент инерции нагрузки,  $k_{тр2}$  – коэффициент трения колеса (нагрузки) о поверхность,  $i$  – передаточное отношение между нагрузкой и валом двигателя (например, если используется редуктор или мультипликатора  $i \neq 1$ , однако в нашем случае примем  $i = 1$  – как будто на валу сидит колесо/цилиндр).

Примем  $k_{тр1} = 0.05$  ;  $k_{тр2} = 0.1$ ;  $J_{нагр} = 10^{(-2)}$  кг·м<sup>2</sup>

Тогда схема примет вид как показано на рисунке 17

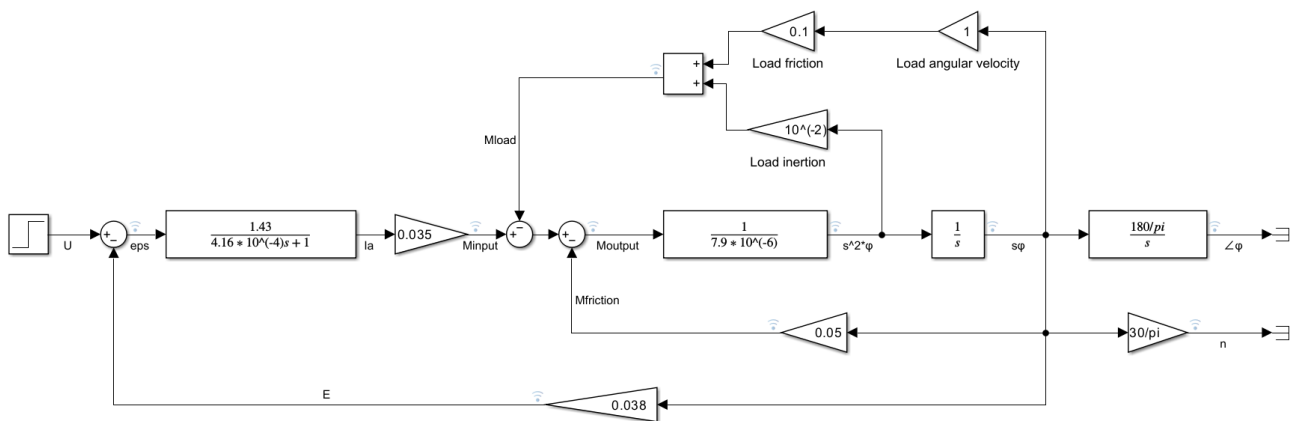


Рис. 17. Структурная схема ДПТ при учете момента нагрузки и момента трения двигателя

Также для удобства чтения графиков, на выходе добавлены функции перевода искомых величин из радиан в градусы, из рад/с в об/мин.

Выведем графики моментов (всех), оборотов в минуту, угла вала, тока якоря (рис. 18)

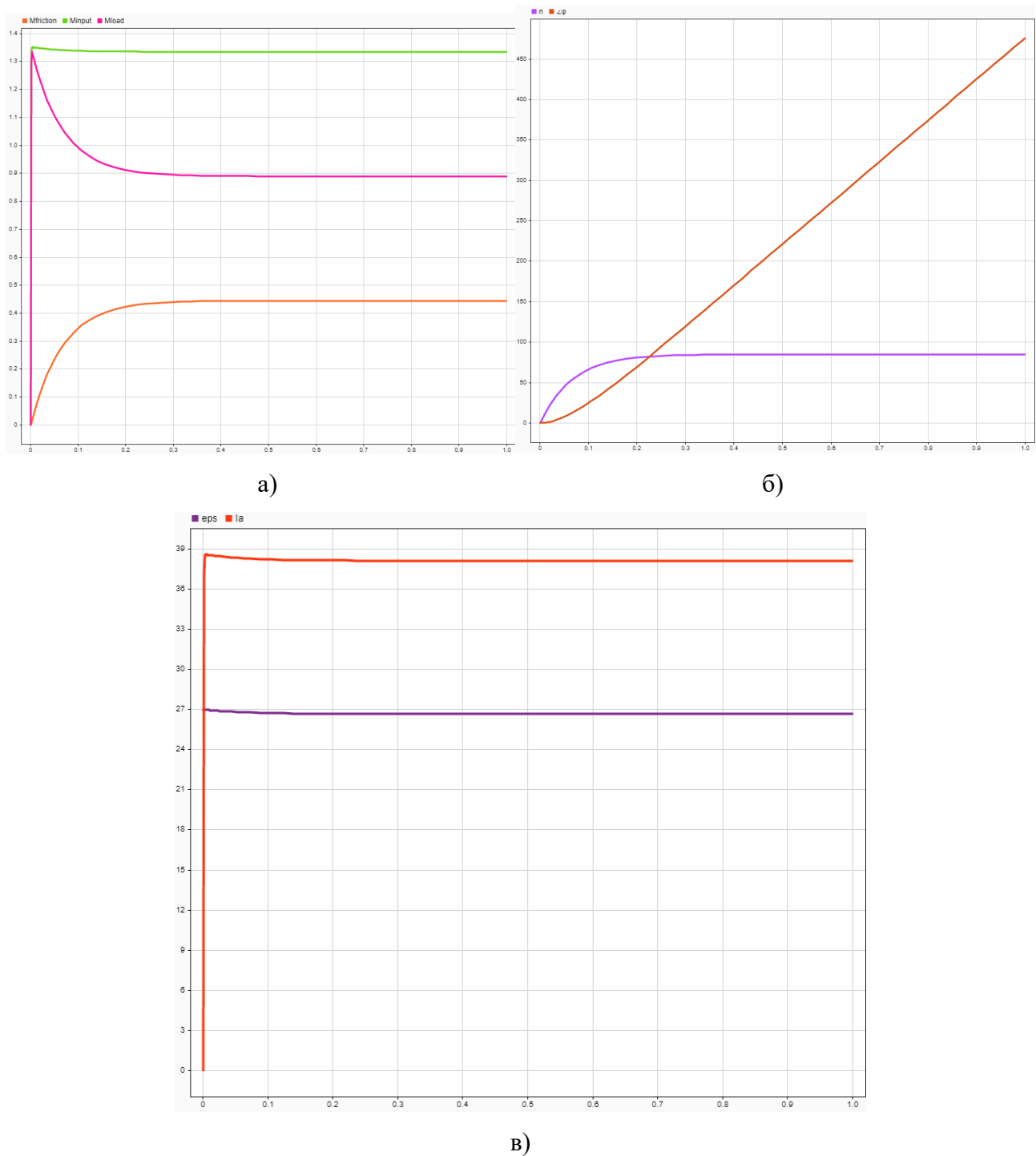


рис. 18 Полученные графики при учете момента нагрузки и момента трения. а – графики моментов; б – графики скорости вращения вала и угла вала двигателя; в – график тока якоря и ошибки

## II. Численное моделирование

В данном пункте стоит задача познакомиться с различными Solver-ами (решателями дифференциальных уравнений первого порядка с входным сигналом системы  $u$  и её состоянием  $x$  (формула 9)) в среде разработки Simulink.

$$\dot{x} = A(t)x + B(t)u; \quad (9)$$

Начнем с солвера ode4 (Метод Рунге Кутты 4го порядка) на примере решения уравнения Ван дер Поля (формула 10). Структурная схема представлена на рисунке 19.

$$\ddot{y} - \mu(1 - y^2)\dot{y} + y = 0; \quad (10)$$

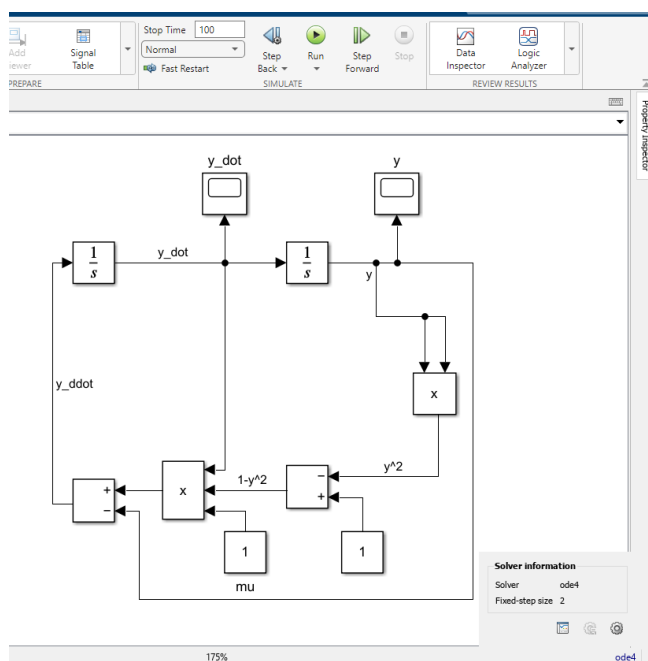
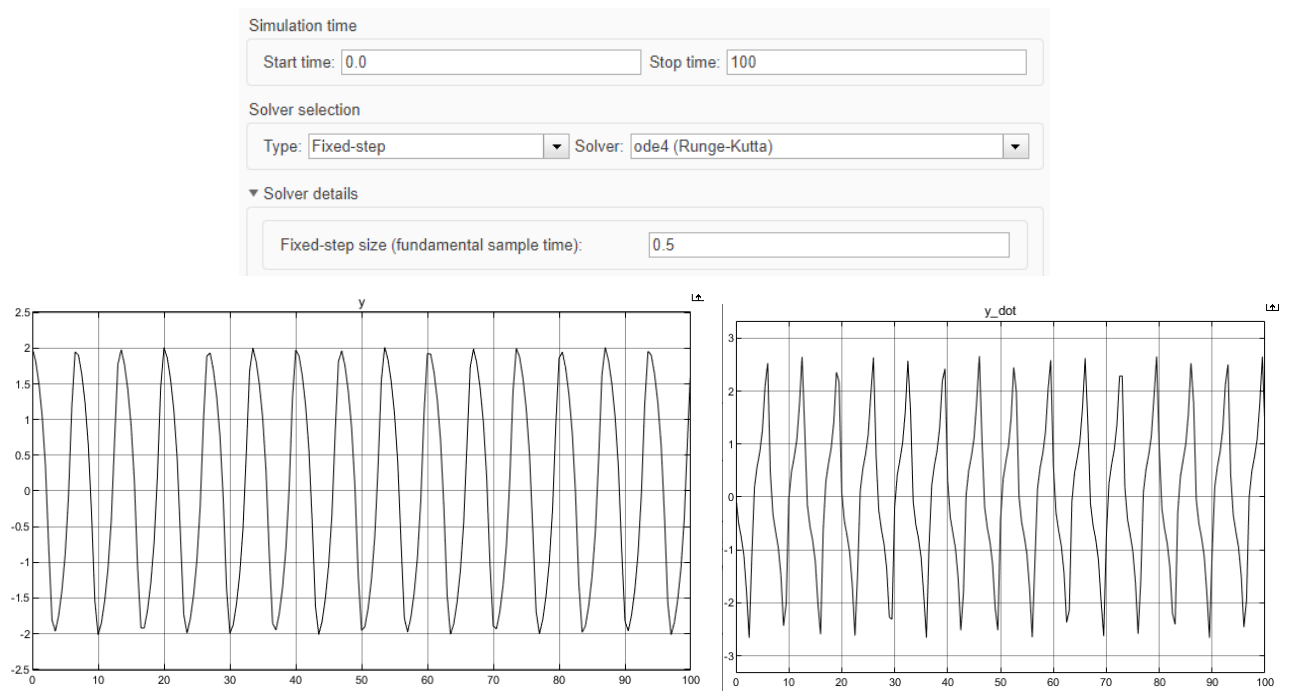


Рис. 19. Солвер ode4 с шагом 2 и временем моделирования 100

На выходе система выдаст ошибку интегрирования на при значении времени  $T = 6$  с. (значение системы не является конечным числом). Понизим шаг до 0.5 и выведем полученные графики (рис. 20). Полученный график 20а показывает решение ЛОДУ, а 20б отображает её производную

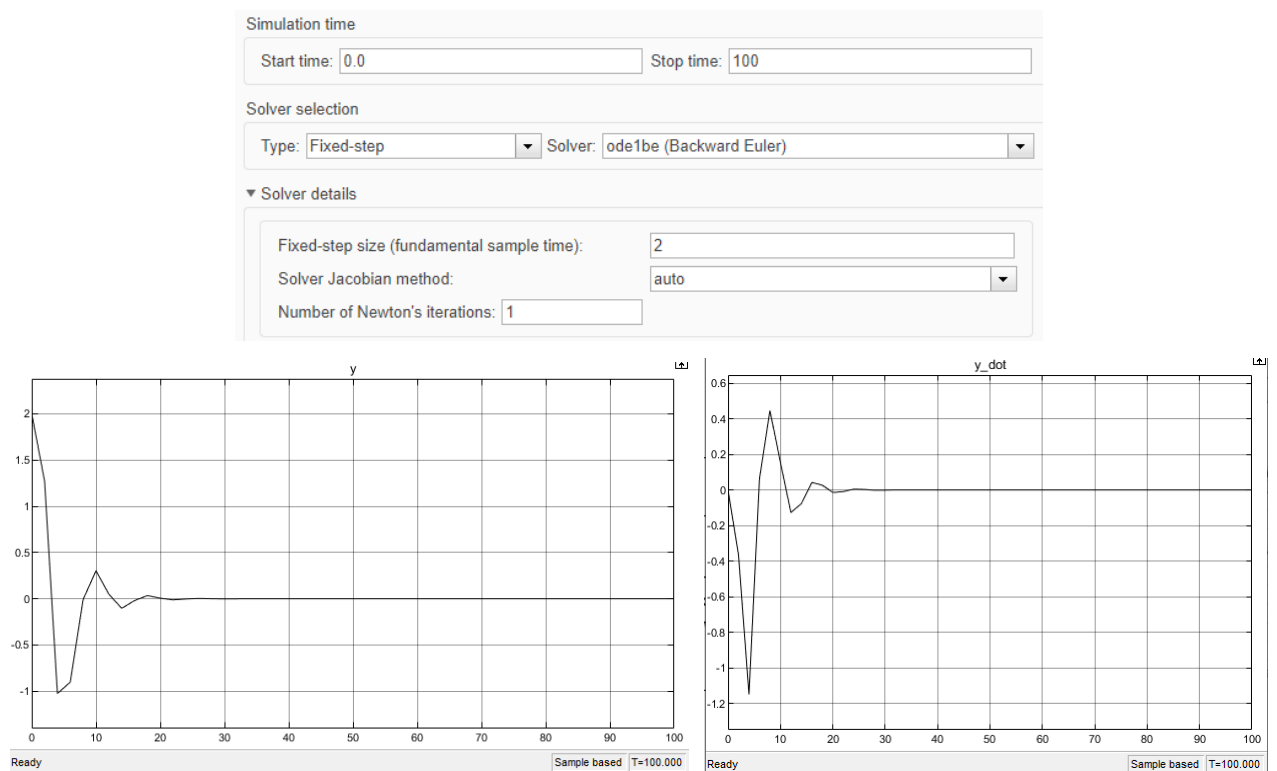
Применим солвер ode1be (Обратный метод Эйлера) с шагами 2 с., 1 с., 0.1 с., 0.01 с., 0.001 с., и выведем полученные графики (рис 21).



a)

б)

Рис. 20. Графики при ode4 и шагом 0.5 ( а – график  $y(t)$ , б – график  $\dot{y}(t)$ )



a)

Simulation time

Start time:  Stop time:

Solver selection

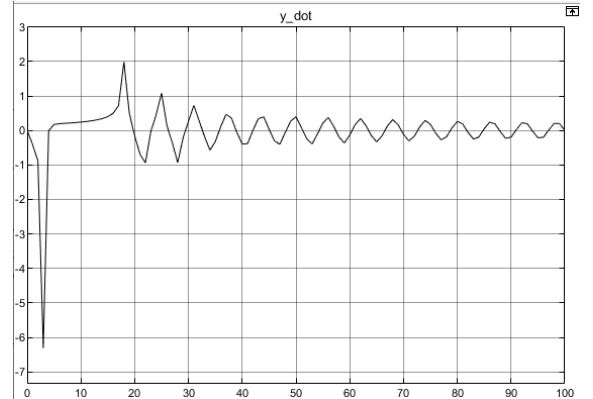
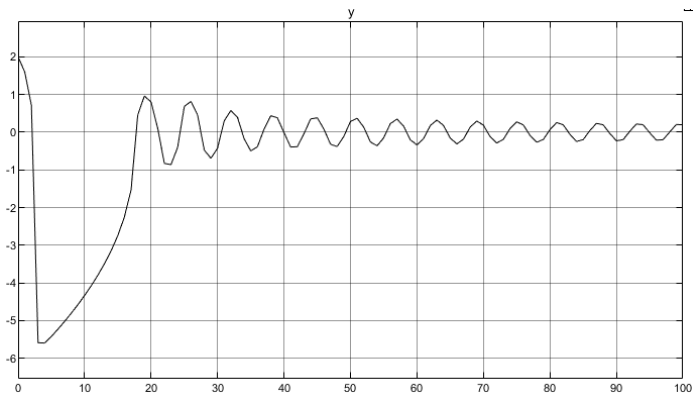
Type:  Solver:

▼ Solver details

Fixed-step size (fundamental sample time):

Solver Jacobian method:

Number of Newton's iterations:



6)

Simulation time

Start time:  Stop time:

Solver selection

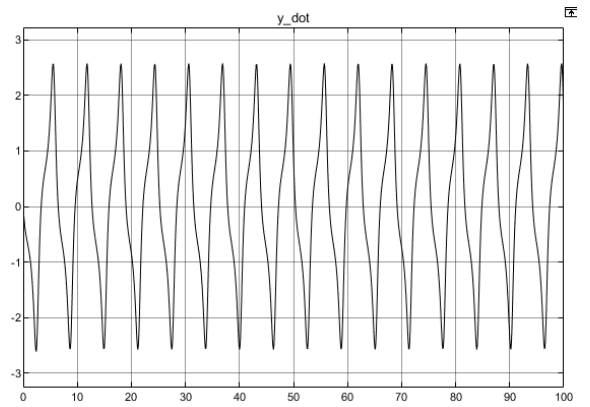
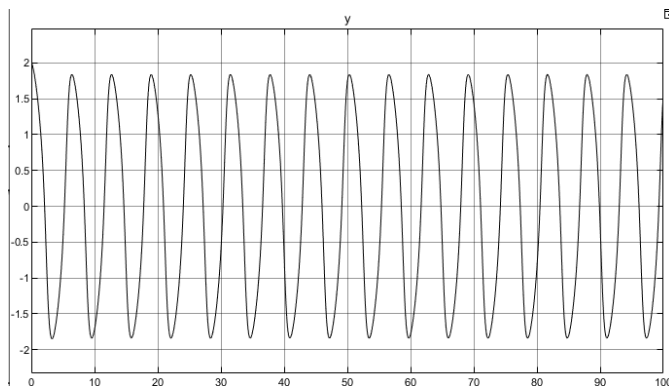
Type:  Solver:

▼ Solver details

Fixed-step size (fundamental sample time):

Solver Jacobian method:

Number of Newton's iterations:



B)

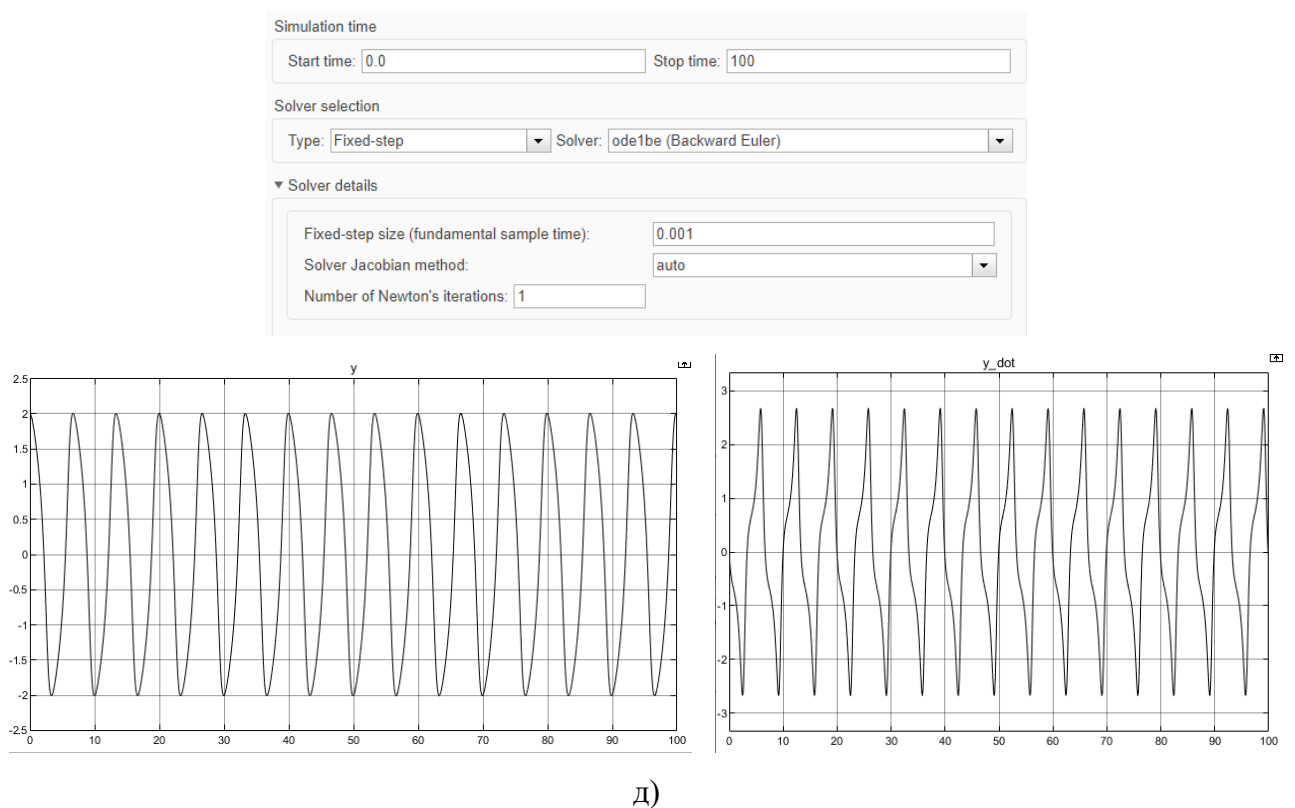
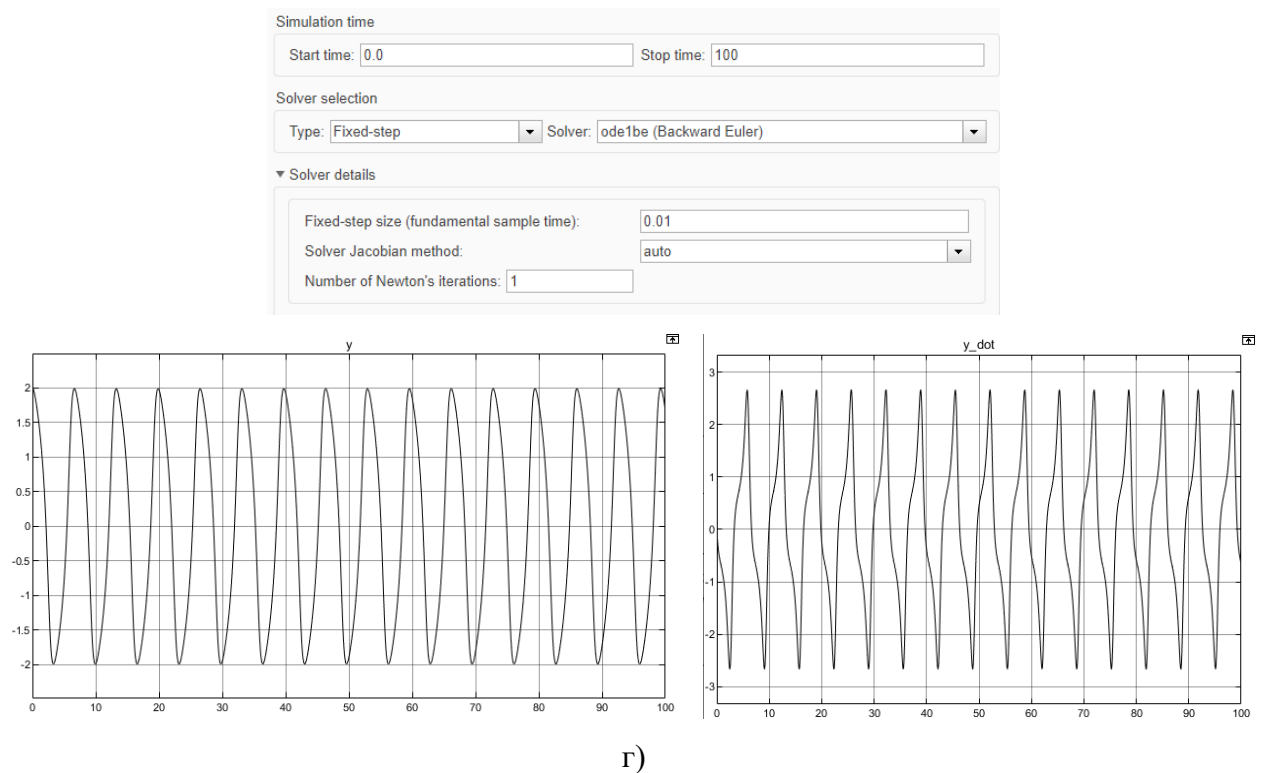
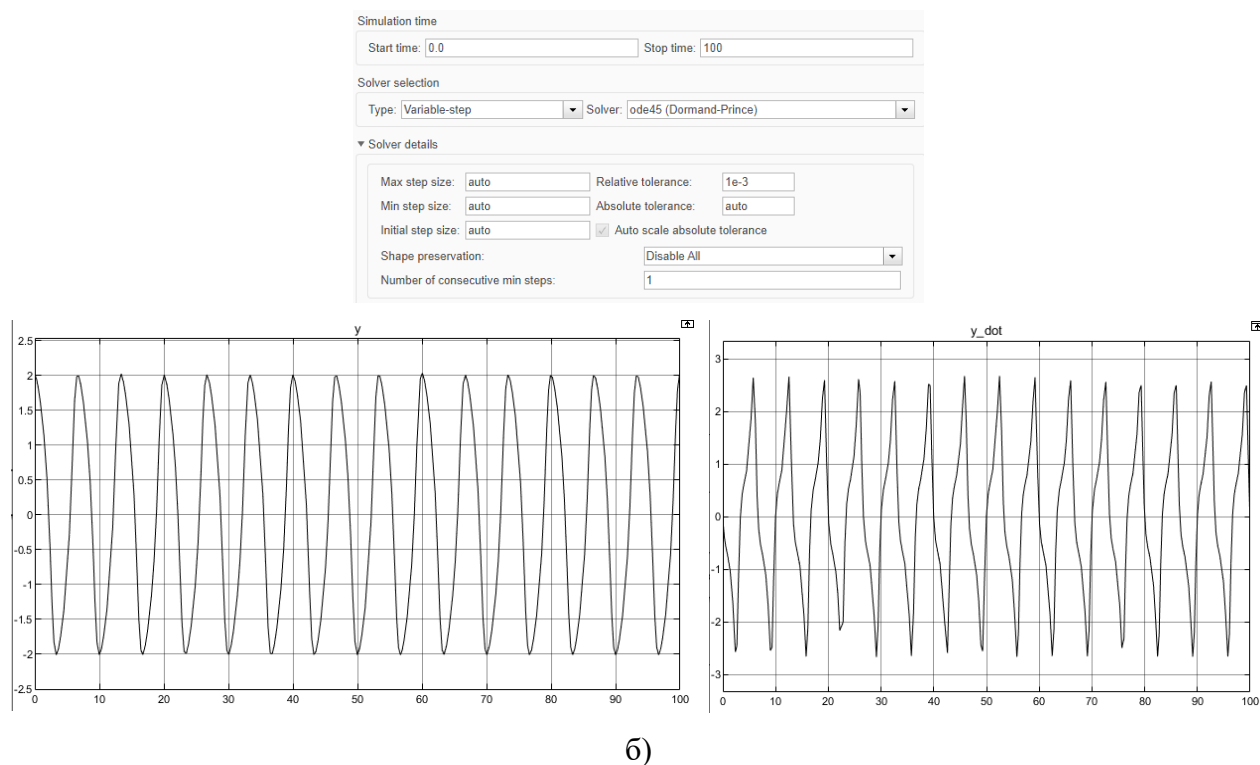
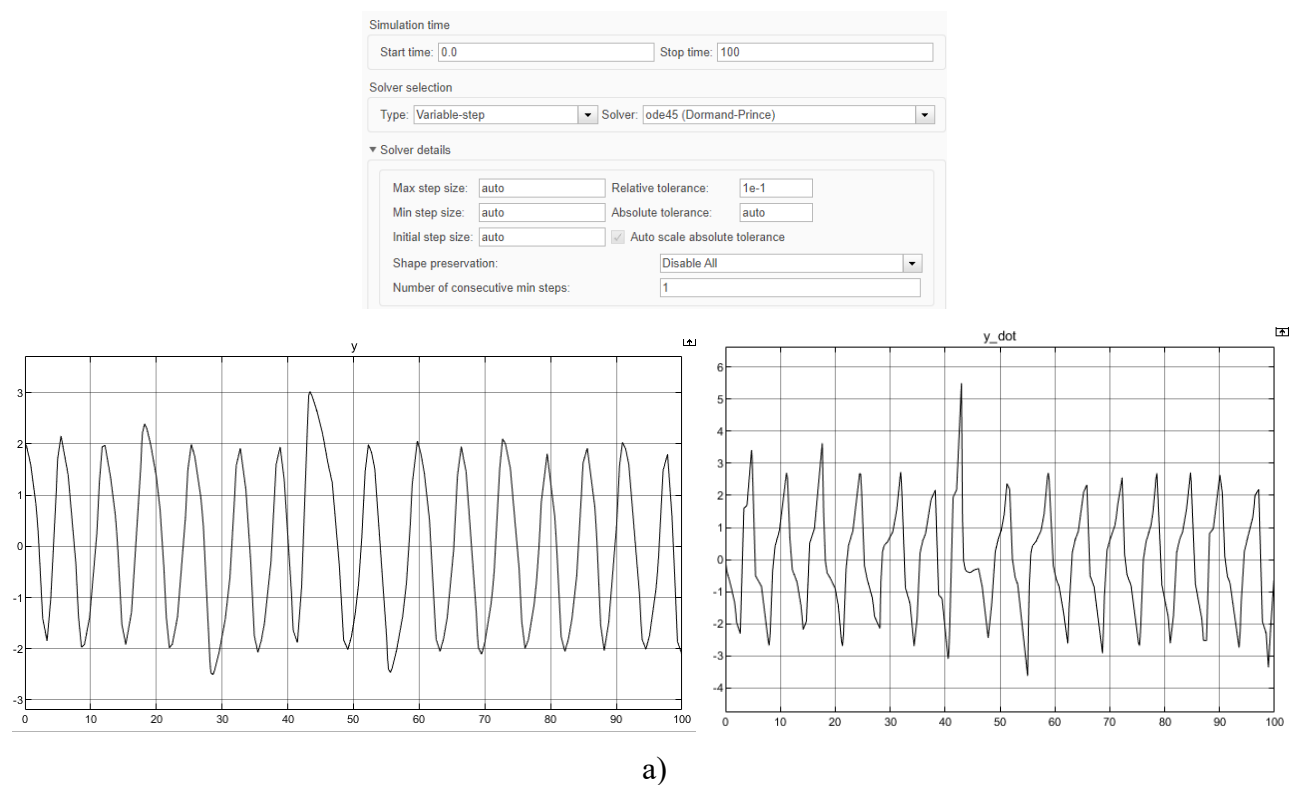


Рис. 21. Графики при ode1be (а – шаг 2с.; б – 1с.; в – 0.1с.; г – 0.01с.; д – 0.001с.)

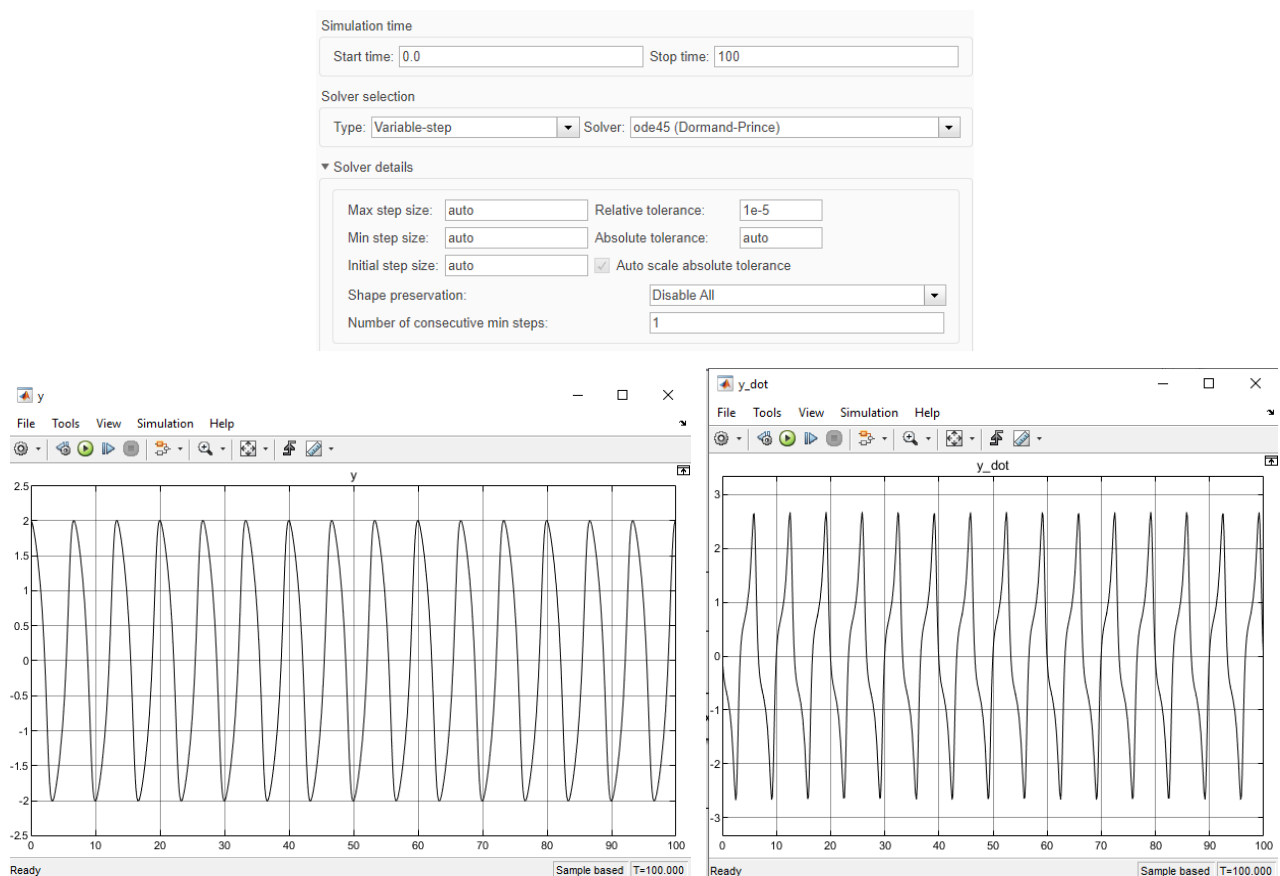
По полученным графикам можно сказать, что уменьшение шага повышает точность решения ЛОДУ, но при этом и увеличивает время работы самой программы. При этом стоит отметить, что после определенного значения шага (в нашем случае 0.1 с.) решения ЛОДУ не сильно отличается от меньших значений

шага, значит для каждого отдельного системы нужно подбирать своё собственное, оптимальное значение шага.

Иным подход - подбор необходимой точности вычислений, с указанием граничных значений шага. Рассмотрим ode45 с переменной длиной шага и параметром Relative tolerance  $1e-1$  (рис. 22а),  $1e-3$  (рис. 22б),  $1e-5$  (рис. 22в).







в)

Рис. 22. Графики с переменным шагом ode45 (а –  $1e-1$ ; б –  $1e-3$ ; в –  $1e-5$ )

Сравнивая графики на рис. 22, можно сказать, что оптимальное значение находится около  $1e-3$ , поскольку график 22б от 22в отличается незначительно, а график с пределом  $1e-1$  уступает двум другим. Такой метод является более оптимальным, чем подбор шага вручную.

Рассмотрим понятие жесткости системы. Для этого изменим параметр  $\mu$  на значение  $1e-7$ , и выведем полученный график производной (солвер ode45 с точностью  $1e-3$ ; время моделирования 100 секунд) (рис. 23)

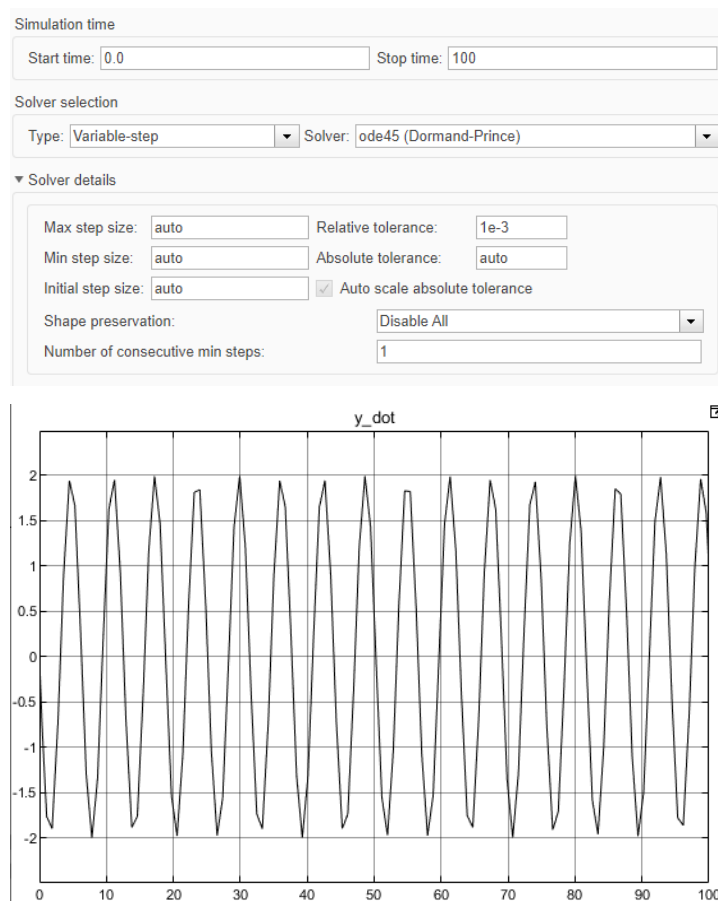


Рис. 23. Пример жесткой системы

График 23, «более гладок» на длинных промежутках, но при этом в точках перегиба имеет скачки значения (образуются «пики», «острые углы»). Система на точках перегибах и иных «ломких» промежутках ведёт грубее.

Для устранения проблемы были разработаны солверы, стремящиеся уменьшить влияние этой ошибки, сделать графики более предсказуемыми в поведении на точках перегибов. Один из них – солвер ode15s, графики системы которого представлены на рисунке 24.

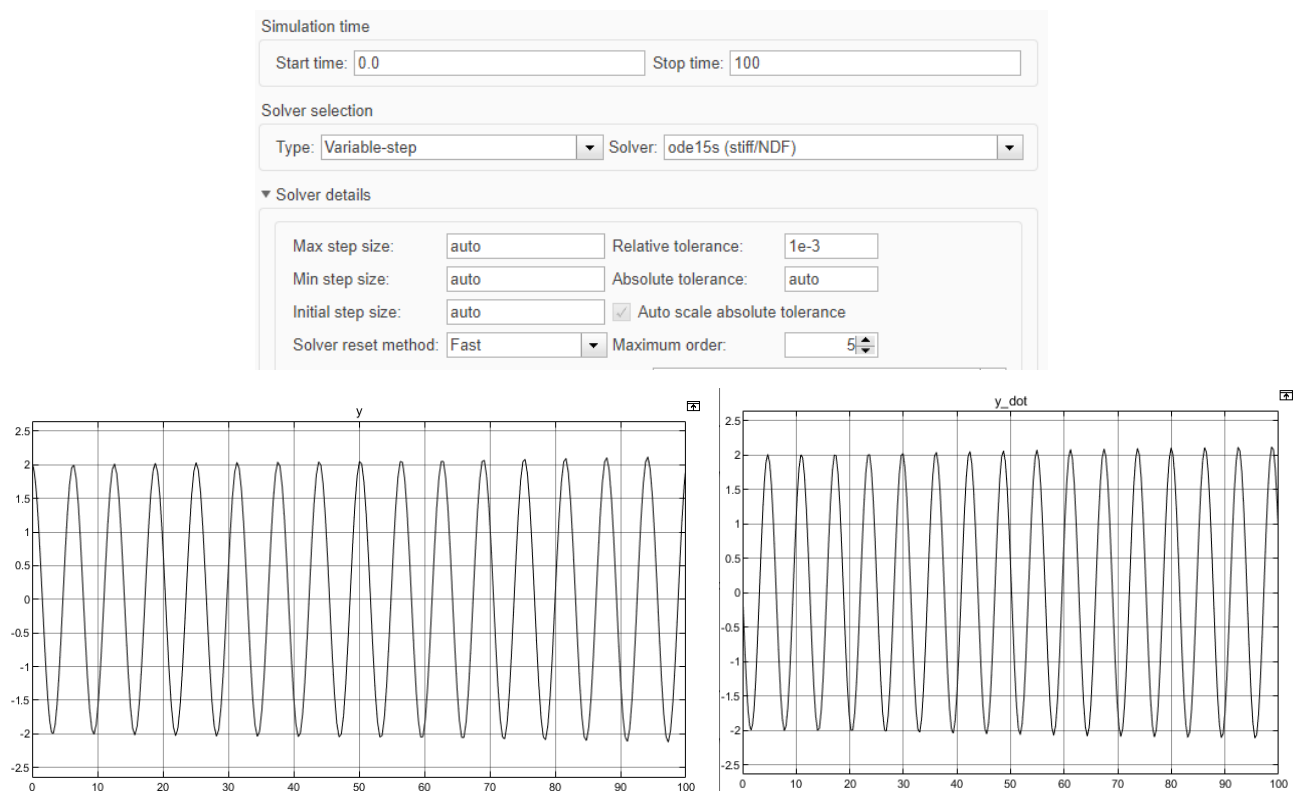


Рис. 24. Пример жесткой системы с солвером ode15s

Видно, что применение указанного солвера *значительно уменьшает «остроту» графика*, делает его более плавным и «спокойным», но не устраняет первоначальную проблему полностью. При том очевидно, что за длительный промежуток моделирования набегает некоторая ошибка, которую также необходимо брать в расчёт.

## Приложение

1. Публичный репозиторий для лабораторных работ по ТАУ // GitHub  
URL: <https://github.com/RiXenGC/Control-Theory.git>
2. Aborisade, David O DC Motor with Load Coupled by Gears Speed Control using Modified Ziegler-Nichols Based PID Tunings // IISTE. - 2014. - №4. - С. 58-67.