|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 № 1**

по курсу «Теория автоматического управления»

Вариант 13

Выполнил: Садовец Роман

Группа: СМ7-62Б

Проверил(а):

Москва, 2024 г.

1. **Знакомство с Git и GitHub**
   1. **Шпаргалка. Базовые команды консоли**
      1. **Навигация**

* **pwd –** текущая директория
* **ls –** файлы и папки в текущей директории
* **ls –a -** файлы и папки в текущей директории, с учетом скрытых
* **cd <dir> -** перейти в папку <dir>
* **cd .. –** перейти на уровень выше (в родительскую папку)
* **cd ~ -** перейти в домашнюю директорию
* **cd / -** перейти в корневую директорию
  + 1. **Работа с файлами и папками**
* **touch <file-name> -** создать файл <file-name> в текущей директории (указывается формат файла)
* **mkdir <name-dir> -** создать директорию в <name-dir> в текущей директории
* **cp <file-name> <location> -** скопируй файл <file-name> в директорию <location>
* **mv <file-name> <location> -** перемести файл <file-name> в директорию <location>
* **cat <file.txt> -** распечатай содержимое текстового файла file.txt.
* **rm <file-name>** - удали файл <file-name>
* **rmdir <dir-name> -** удали пустую папку <dir-name>
* **rm –r <dir-name> -** удали папку <dir-name> и все её содержимое
  + 1. **Полезные возможности**
* **--help –** вывести всю дополнительную информацию о команде
* Команды необязательно печатать и выполнять по очереди. Можно указать их списком — разделить двумя амперсандами (**&&**).
* У консоли есть собственная память — **буфер** с несколькими последними командами. По ним можно перемещаться с помощью клавиш со стрелками вверх (**↑**) и вниз (**↓**).
* Чтобы не вводить название файла или папки полностью, можно набрать первые символы имени и дважды нажать Tab. Если файл или папка есть в текущей директории, командная строка допишет путь сама.
* Например, вы находитесь в папке dev. Начните вводить cd first и дважды нажмите Tab. Если папка first-project есть внутри dev, командная строка автоматически подставит её имя. Останется только нажать Enter.
  1. **Шпаргалка. Начало работы с Git**
     1. **Возможности**
* **git init** — инициализируй репозиторий.
* **git push** — загрузи коммиты в удалённый репозиторий после того, как он был привязан с помощью флага -u.
* **git add todo.txt** — подготовь файл todo.txt к коммиту;
* **git add –all** — подготовь к коммиту сразу все файлы, в которых были изменения, и все новые файлы;
* **git add .** — подготовь к коммиту текущую папку и все файлы в ней.
* **git commit -m "Комментарий к коммиту."** — сделай коммит и оставь комментарий, чтобы было проще понять, какие изменения сделаны;
* **git push** — добавь изменения в удалённый репозиторий.
* **git log**— выведи подробную историю коммитов;
* **git log --oneline** — покажи краткую информацию о коммитах: сокращённый хеш и сообщение.
* **git status** — покажи текущее состояние репозитория.
* **git commit --amend --no-edit** — добавь изменения к последнему коммиту и оставь сообщение прежним;
* **git commit --amend -m "Новое сообщение"** — измени сообщение к последнему коммиту на Новое сообщение.
* **git restore --staged hello.txt** — переведи файл hello.txt из состояния staged обратно в untracked или modified;
* **git restore hello.txt** — верни файл hello.txt к последней версии, которая была сохранена через git commit или git add;
* **git reset --hard b576d89**— удали все незакоммиченные изменения из staging и «рабочей зоны» вплоть до указанного коммита.
* **git diff** — покажи изменения в «рабочей зоне», то есть в modified-файлах;
* **git diff a9928ab 11bada1** — выведи разницу между двумя коммитами;
* **git diff –staged** - покажи изменения, которые добавлены в staged-файлах.
  + 1. **Связь локального и удаленного Git-а**
* **git remote add origin** <https://github.com/YandexPracticum/first-project.git> — привяжи локальный репозиторий к удалённому с URL https://github.com/YandexPracticum/first-project.git;
* **git remote -v** — проверь, что репозитории действительно связались;
* **git push -u origin main** — в первый раз загрузи все коммиты из локального репозитория в удалённый с названием origin.
* Также для адекватной работы необходимо использование SSH-ключа. Если ранее не создавался, то для генерации: **ssh-keygen -t ed25519 -C "%FirstName% %LastName% (%e-mail%) %HostName%"** и потом чтение внутренностей **cat ~/.ssh/id\_ed25519.pub.** Подробнее – в отдельном файле в директории
  1. **Шпаргалка. Основы работы с ветками Git**
* **git clone** [**git@github.com:YandexPraktikum/first-project.git**](mailto:git@github.com:YandexPraktikum/first-project.git) — склонируй репозиторий с URL first-project.git из аккаунта YandexPraktikum на мой локальный компьютер.
* **git branch feature/the-finest-branch**— создай ветку от текущей с названием feature/the-finest-branch;
* **git checkout -b feature/the-finest-branch** — создай ветку feature/the-finest-branch и сразу переключись на неё.
* **git branch** — покажи, какие есть ветки в репозитории и в какой из них я нахожусь (текущая ветка будет отмечена символом \*);
* **git branch -a** — покажи все известные ветки, как локальные (в локальном репозитории), так и удалённые (в origin, или на GitHub).
* **git checkout feature/br** — переключись на ветку feature/br.
* **git branch -d br-name** — удали ветку br-name, но только если она является частью main;
* **git branch -D br-name** — удали ветку br-name, даже если она не объединена с main.
* **git merge dev** — объедини ветку dev с текущей активной веткой.
* **git push -u origin my-branch** — отправь новую ветку my-branch в удалённый репозиторий и свяжи локальную ветку с удалённой, чтобы при дополнительных коммитах можно было писать просто git push без -u;
* **git push my-branch** — отправь дополнительные изменения в ветку my-branch, которая уже существует в удалённом репозитории;
* **git pull** — подтяни изменения текущей ветки из удалённого репозитория.

1. **Построение модели ДПТ с ПВ**

Всю разработку будем вести совместно с системой контроля версий git. Ссылка на публичный репозиторий представлена в списке литературы 1.

Начнём с создания директории, в которой будем работать, инициализации системы git в репозитории, а также связи удаленного репозитория с локальным. Также в коде ниже представлена генерация SSH-ключа для связи локального Git-а с удаленным на GitHub (для анонимности персональные данные отсутствуют). Ранее SSH-ключи на компьютере не создавались. Все изменения и разработку будем вести в ветке dev

|  |
| --- |
| git init  mkdir Lab-1  ssh-keygen -t ed25519 -C "%FirstName% %LastName% (%e-mail%) %HostName%"  cat ~/.ssh/id\_ed25519.pub  git config --global user.email "r.sadovec0403@yandex.ru"  git config --global user.name "Roman-RiXenGC"  git config --global init.defaultBranch main  git config --global core.autocrlf true  git config --global core.safecrlf warn  git remote add origin <https://github.com/RiXenGC/Theory-of-Automatic-Control.git>  git checkout -b dev  cd Lab-1 |

Модель двигателя постоянного с последовательным возбуждением (ДПТ с ПВ) описывается следующей системой уравнений:

(1)

где – момент инерции вала двигателя; – угловое положение вала двигателя; – полезный момент, создаваемый двигателем; – момент трения; – сила тока в цепи якоря (ротора) двигателя; , – активное сопротивление и индуктивность обмоток якоря соответственно; – индуктивность обмотки возбуждения; – напряжение управления двигателя по цепи якоря; – противоэдс, наводящееся в цепи якоря; Φ – магнитный поток, создаваемый обмоткой возбуждения; , – конструктивные коэффициенты двигателя;

Примем:

= 7 ∙ 10-6 [кг ∙ м2]

= 1.3 []

= 3.15 []

= 0.0015 [Гн]

= 8 [Ом]

= 0.013 [Гн]

Запишем основные константы в файл constants.m, а также разобьем уравнения системы (1) на подсистемы (рис. 1, 2, 3) и объединим их в одну общую систему (рис. 4), описывающие модель ДПТ с ПВ.

|  |
| --- |
| J\_m = 7 \* 10^(-6); % Angular moment of motor inertion, kg \* m^2  C\_M = 1.3; % Construction coef of motor, N \* m / ( A \* Gn^2 )  C\_omega = 3.15; % Construction coef of motor, V \* s / ( A \* Gn )  L\_a = 0.0015; % Anchor inductivity, Gn  R\_a = 8; % Anchor resistance, Om  L\_ew = 0.013; % Inductivity of excitation winding, Gn  %Viscous friction  Betta = 0.001; %coef of viscous friction |

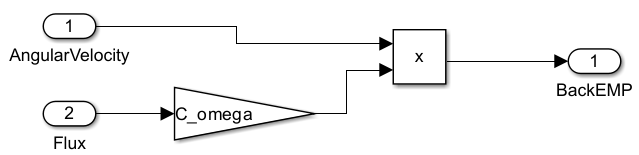


Рис 1. Структурная схема уравнения противоЭДС (уравнения 1.3, 1.6)

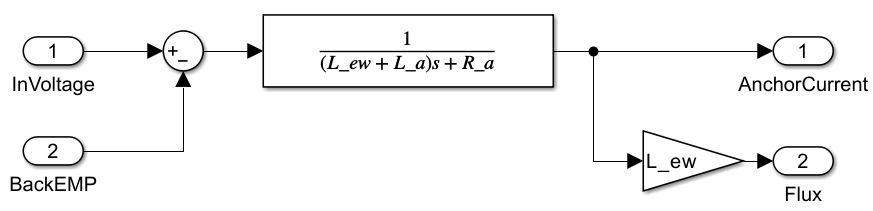


Рис 2. Структурная схема уравнения тока якоря и магнитного потока обмоток двигателя (уравнения 1.4, 1.5)

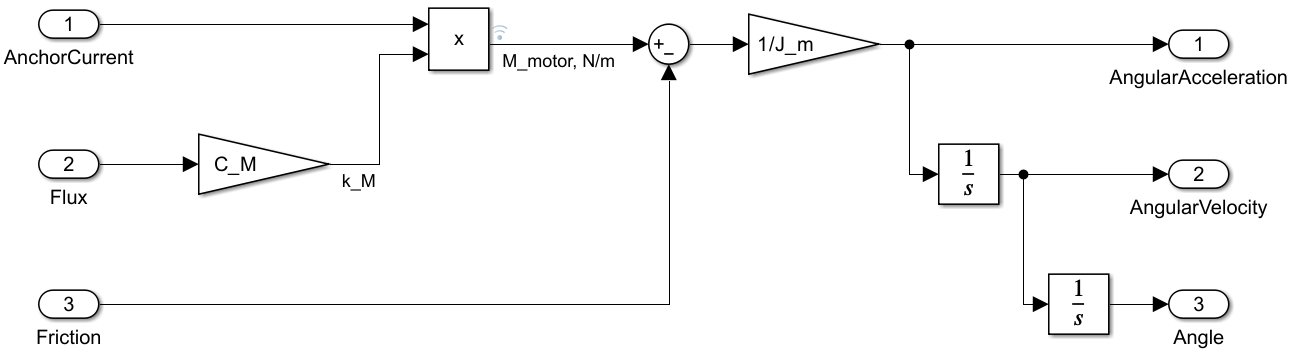
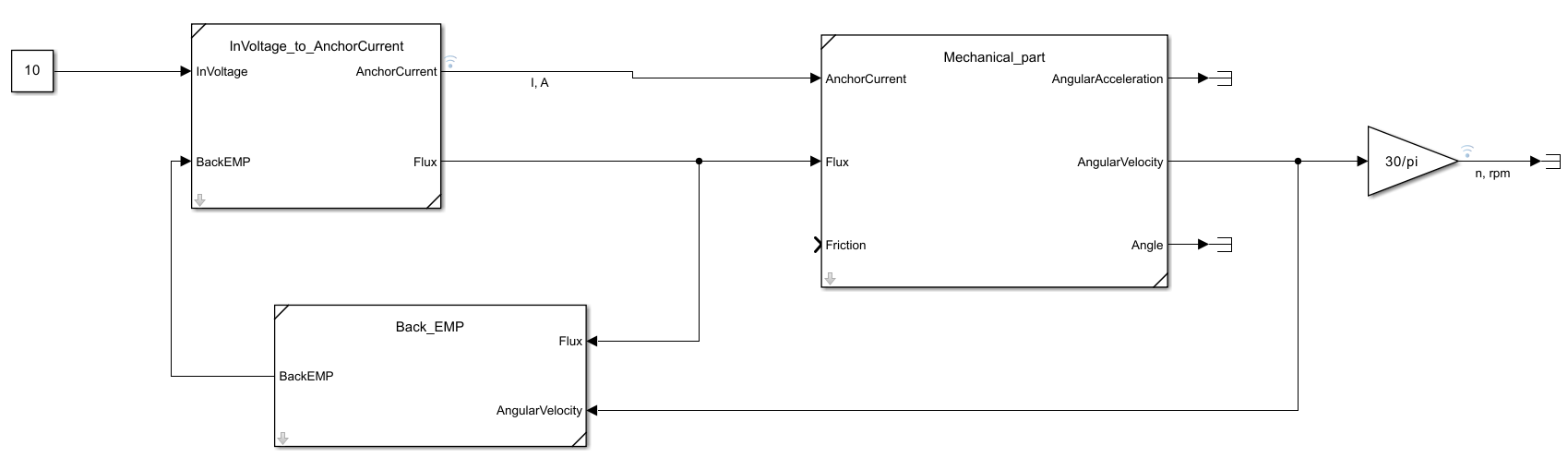


Рис 3. Структурная схема уравнения механической части (уравнения 1.1, 1.2, 1.7)

После получения системы без учета вязкого трения, выведем графики основных технических параметров: тока якоря, момента двигателя, скорости вращения вала (в об/мин), а также механическую характеристику модели при = 5 В (рис. 5, 6) и при = 10 В (рис. 7, 8).

 Рис 4. Структурная схема ДПТ с ПВ без учета вязкого трения

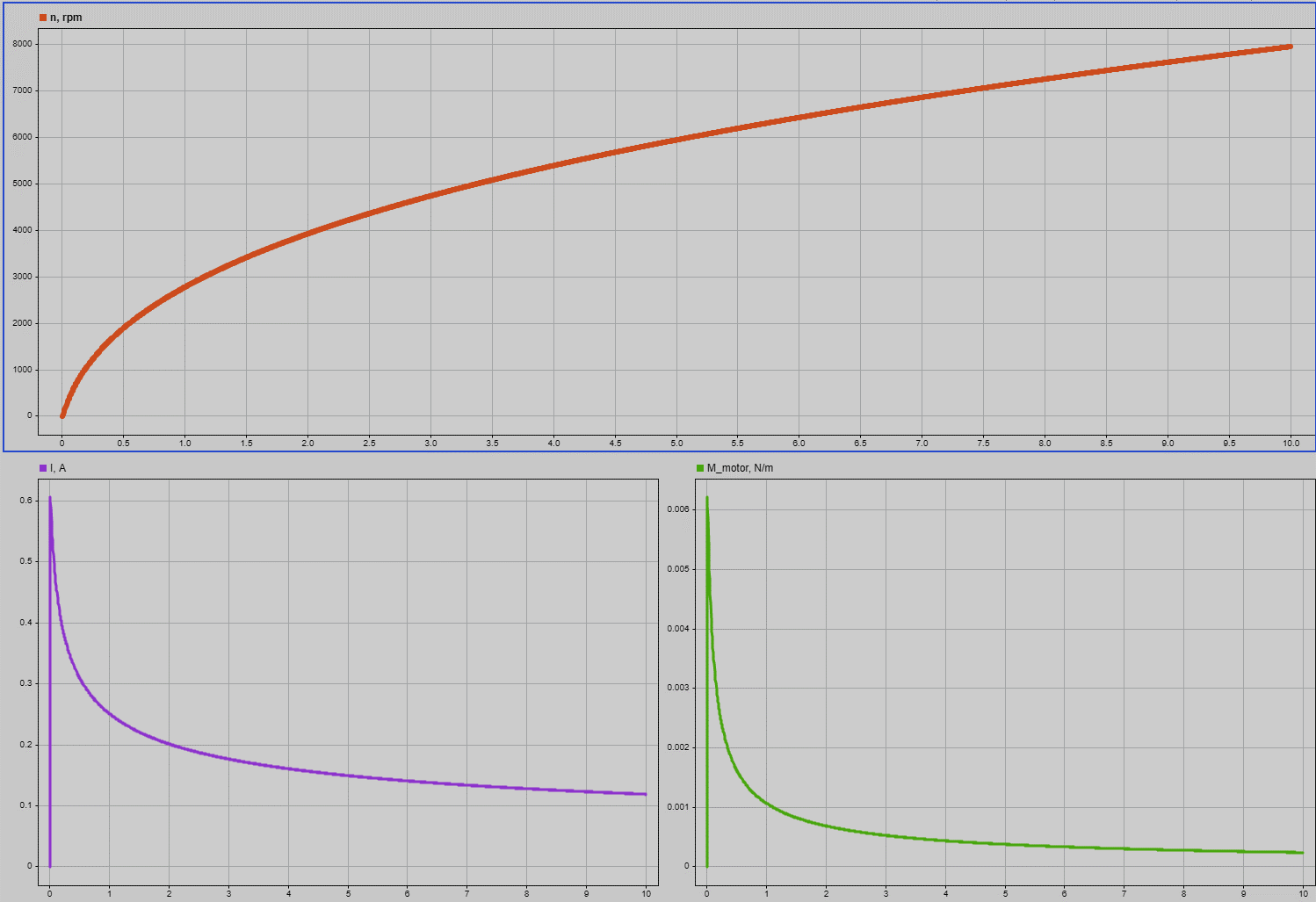


Рис 5. Графики оборотов вала n [об/мин], тока якоря i [A] и полезного момента на двигателе [Н\*м] без учета вязкого трения при подаче на вход

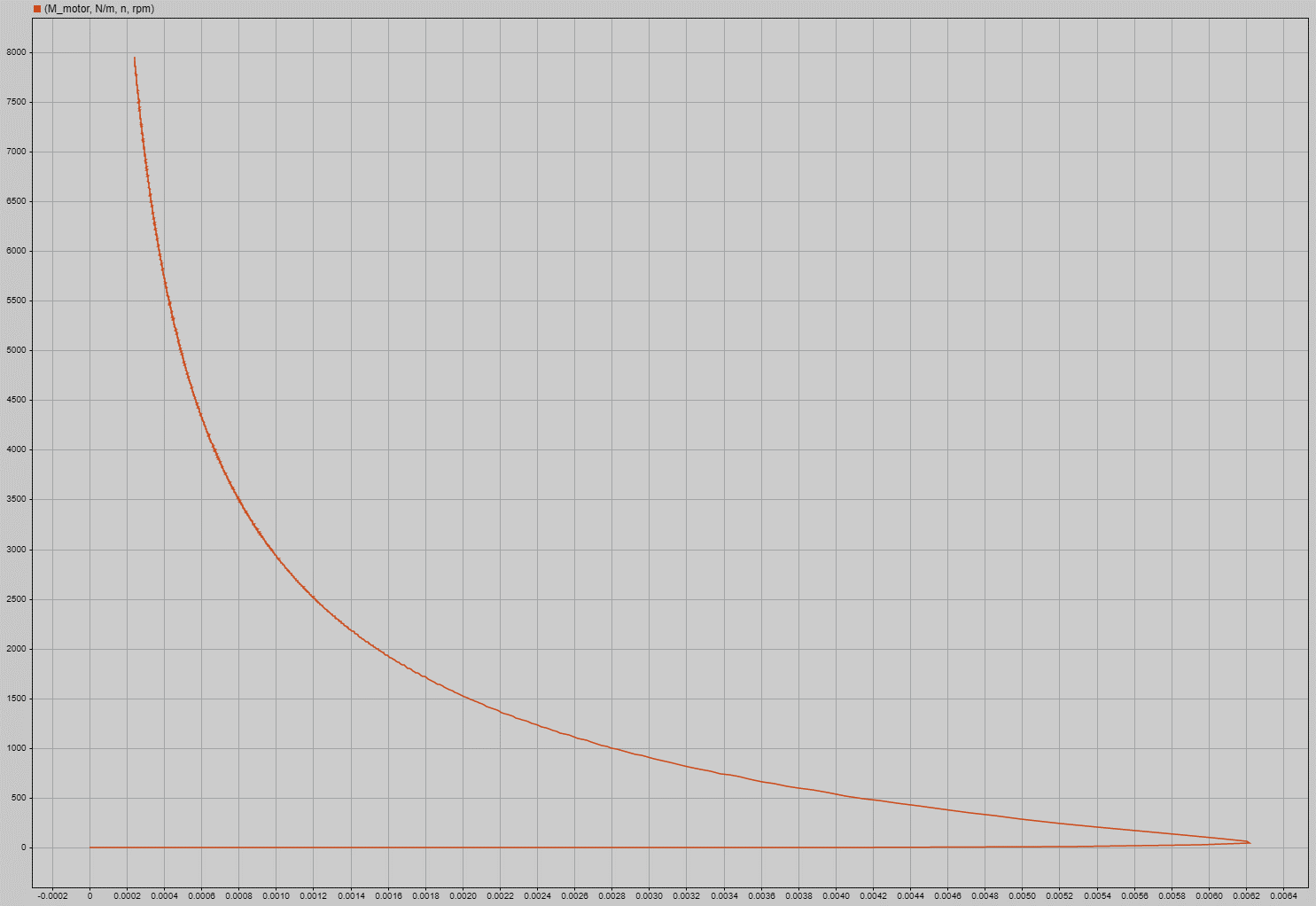


Рис 6. Механическая характеристика ДПТ с ПВ без учета вязкого трения при подаче на вход

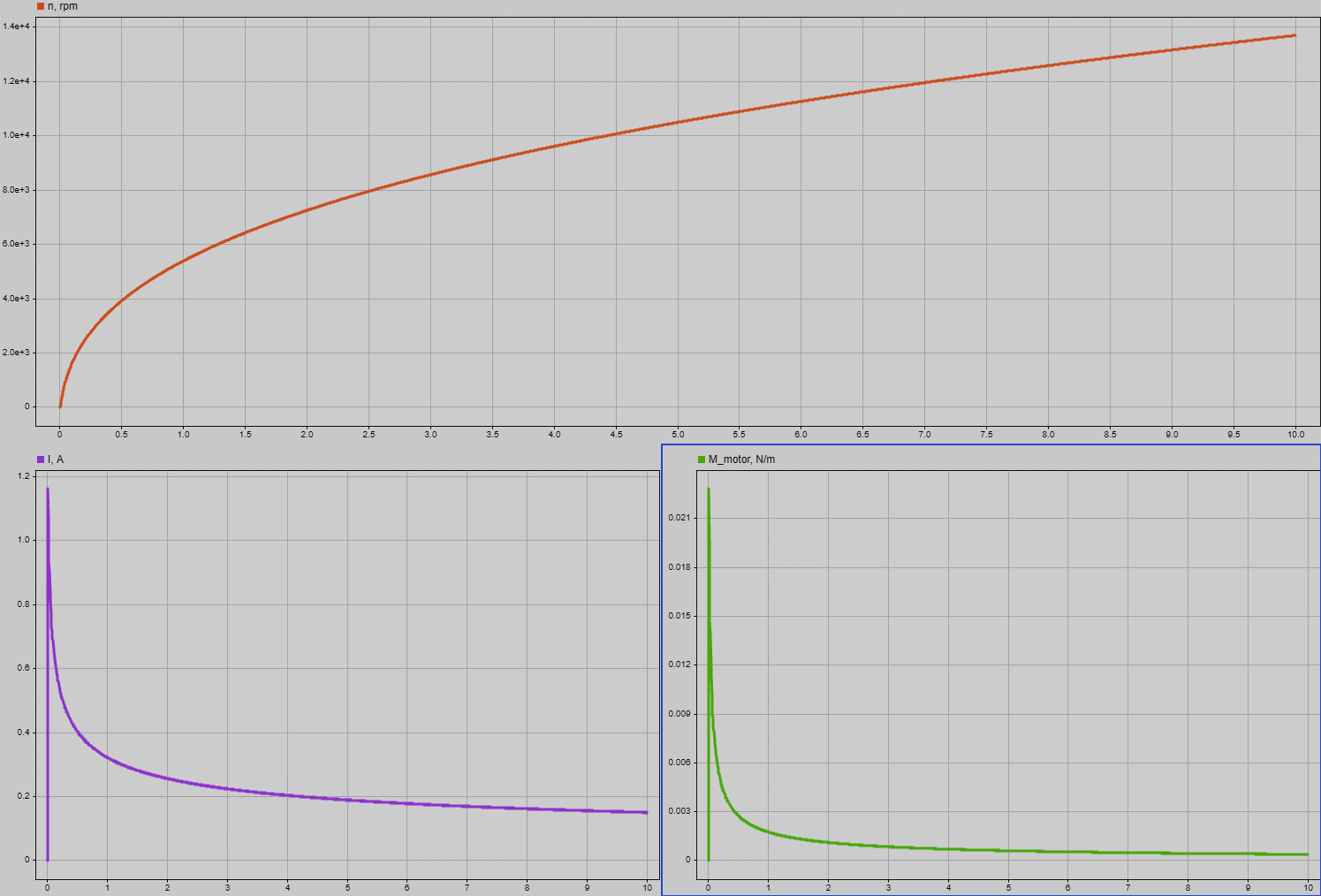


Рис 7. Графики оборотов вала n [об/мин], тока якоря i [A] и полезного момента на двигателе [Н\*м] без учета вязкого трения при подаче на вход

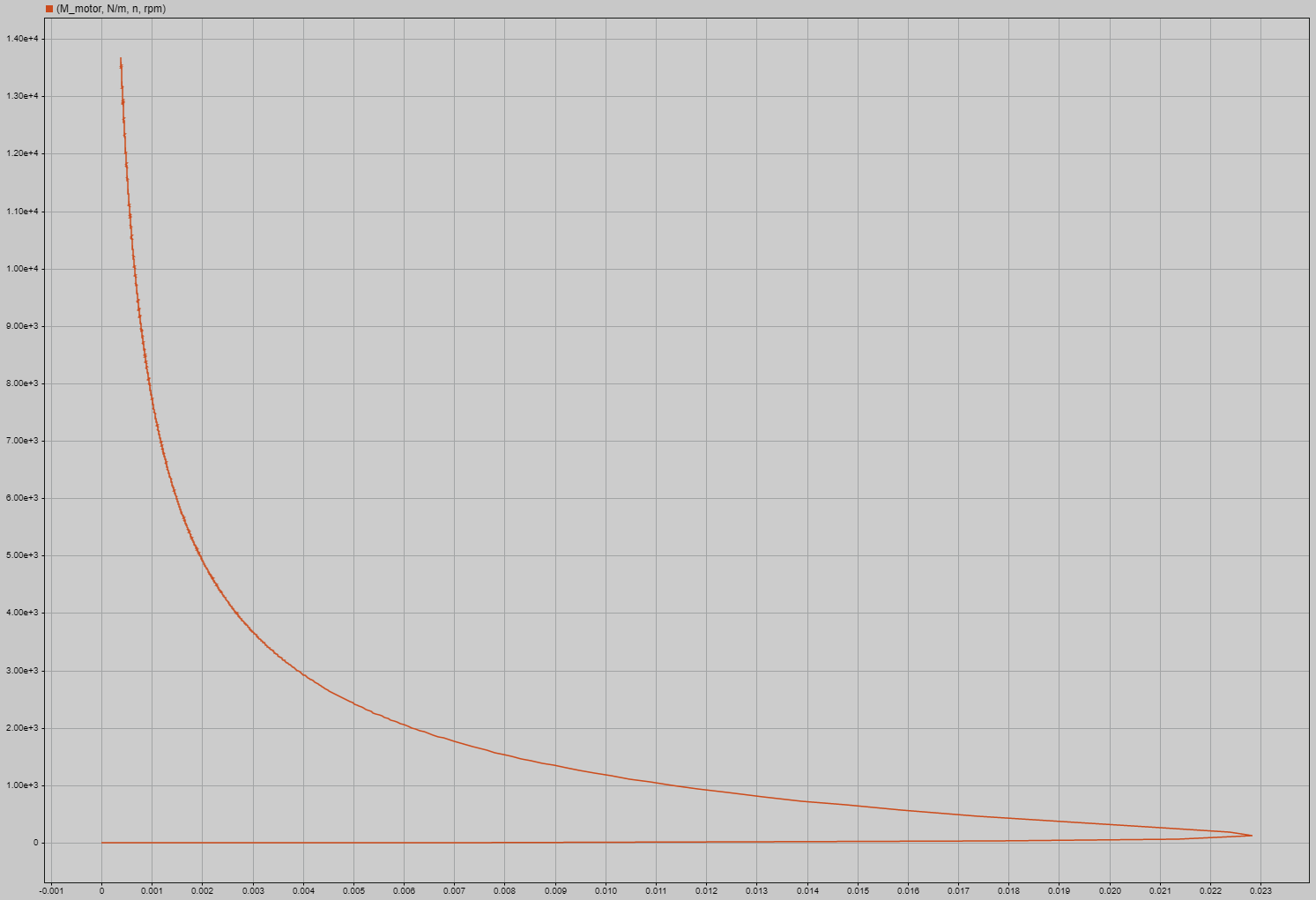


Рис 8. Механическая характеристика ДПТ с ПВ без учета вязкого трения при подаче на вход

На основе полученных графиков можно сказать, что ДПТ с ПВ является нелинейной системой. Вытекает это из отсутствия действия принципа суперпозиции: при подаче на вход 10 В и 5 В графики отличаются не в два раза, а на иную величину. Это говорит о наличии дифференциальных уравнений/нелинейностей в описании модели, т.е. модель ДПТ нелинейна.

Сделаем commit для всей проведенной работы

|  |
| --- |
| git status  git add .  git commit -m "Model of DC motor with serial actuatuon without friction" |

Далее рассмотрим случай при наличии вязкого трения с модели ДПТ с ПВ. Структурная схема с вязким трением представлена на рисунке 9. Введём дополнительную подсистему, описывающую вязкое трения (рис. 10) и выведем графики, аналогичные предыдущим, а именно механическую характеристику (рис. 11) угловую скорость, полезный момент на двигателе, ток якоря (рис.12).

Работу будем проводить в отдельной ветке, выходящей из dev

|  |
| --- |
| git checkout -b feature-viscous\_friction dev |

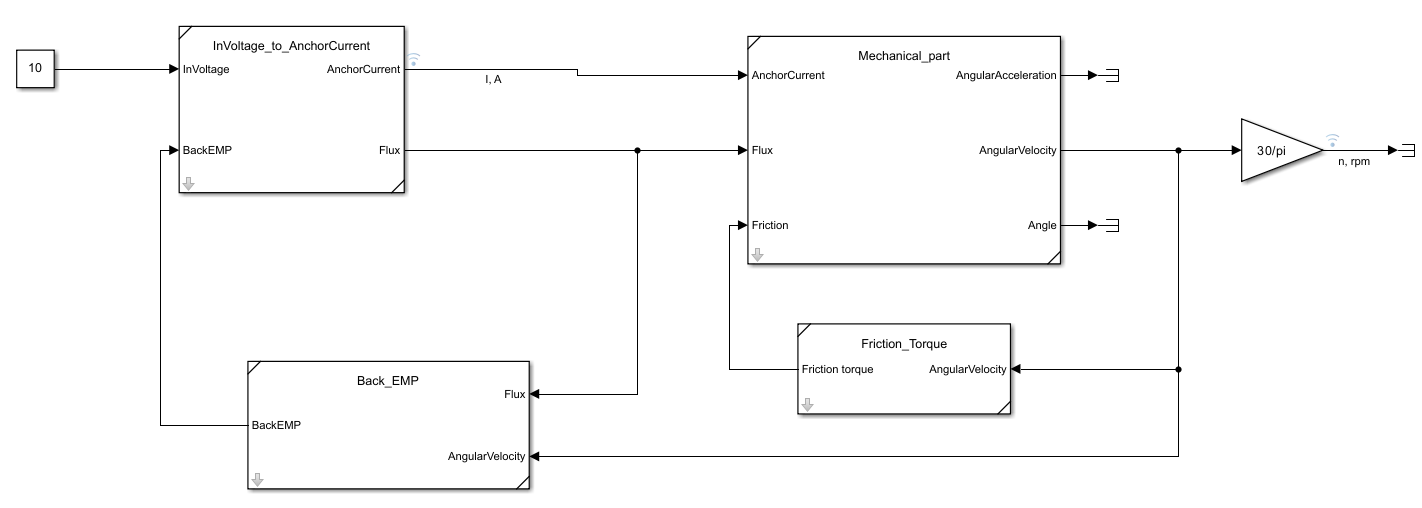


Рис 9. Структурная схема ДПТ с ПВ с учетом вязкого трения

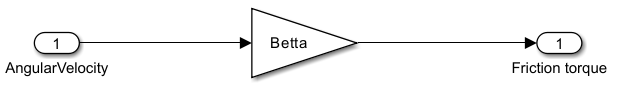


Рис 10. Структурная схема уравнения для вязкого трения

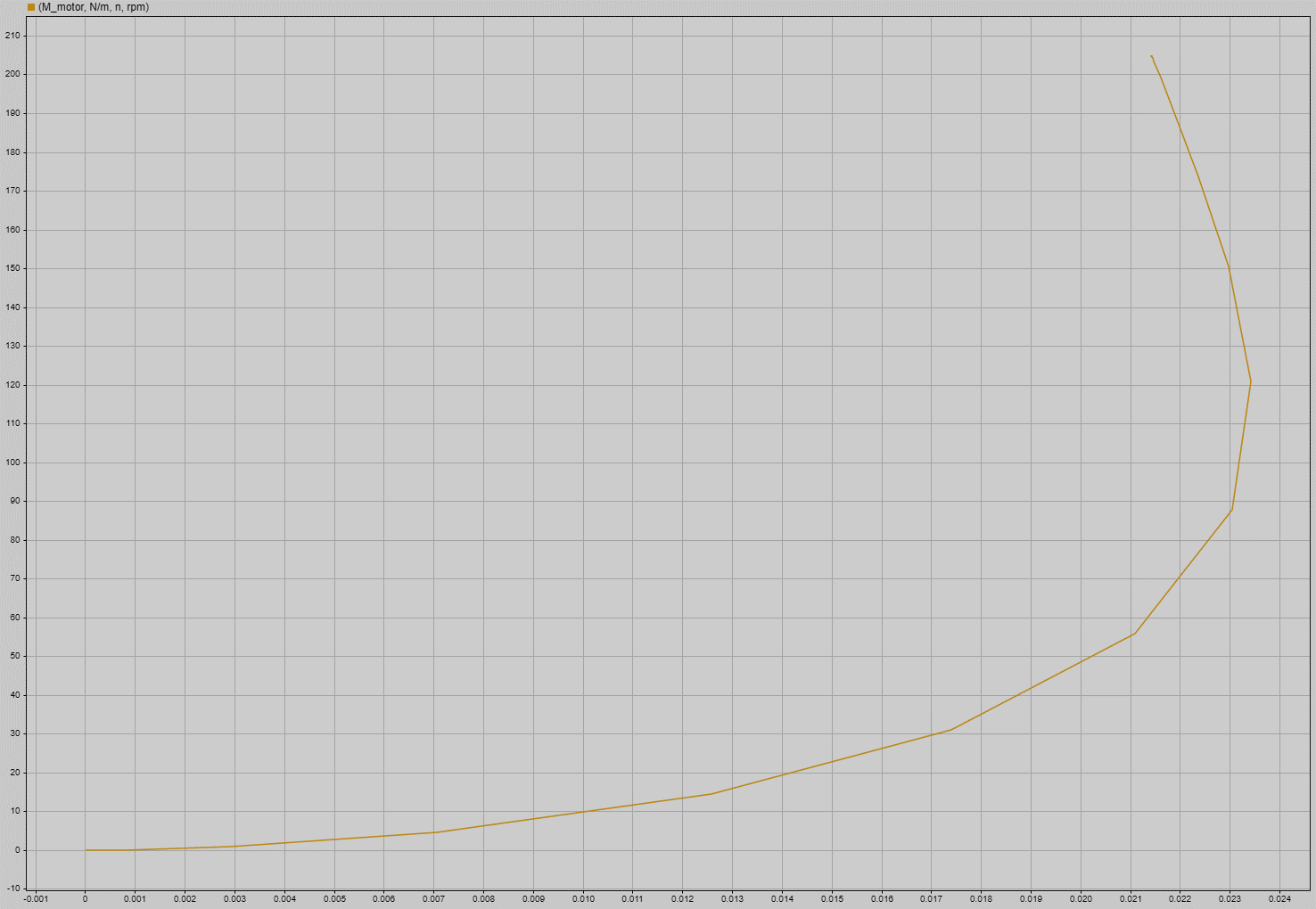


Рис 11. Механическая характеристика ДПТ с ПВ с учетом вязкого трения при подаче на вход

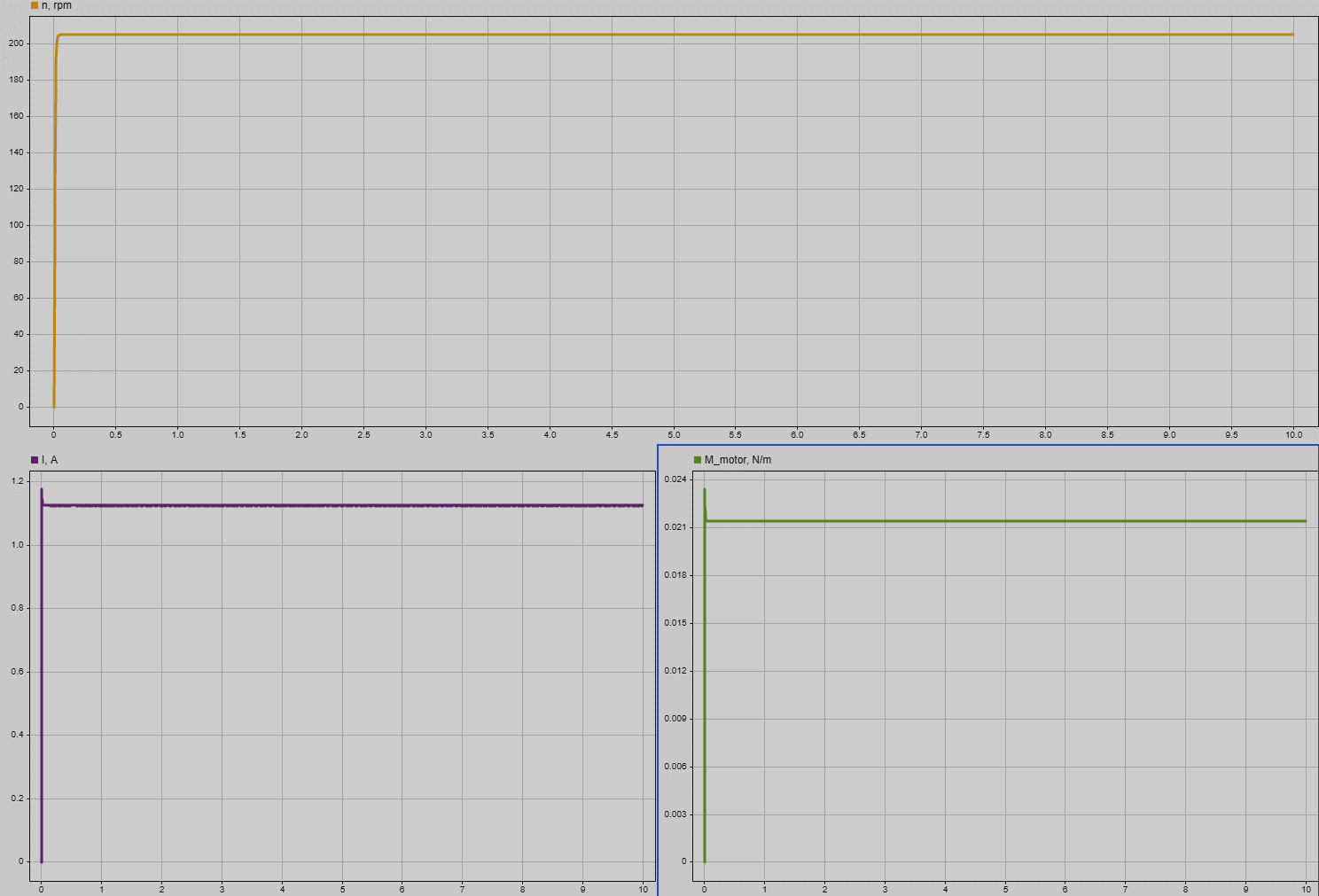


Рис 12. Графики оборотов вала n [об/мин], тока якоря i [A] и полезного момента на двигателе [Н\*м] с учетом вязкого трения при подаче на вход

Полученные результаты в ветке feature-viscous\_friction мержим с веткой dev и пушим её на GitHub

|  |
| --- |
| git status  git add .  git commit -m "Added a viscous friction in model of DC Motor"  git checkout dev  git merge feature-viscous\_friction  git push origin dev |

Далее уже на GitHub делаем pull request ветки dev и мержим её с веткой main.

**Список литературы**

1. Публичный репозиторий для лабораторных по ТАУ // GitHub URL: <https://github.com/RiXenGC/Theory-of-Automatic-Control>