



ЭКСПОНЕНТА

ЦЕНТР ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ
И МОДЕЛИРОВАНИЯ



Как физическое моделирование ускоряет разработку систем управления

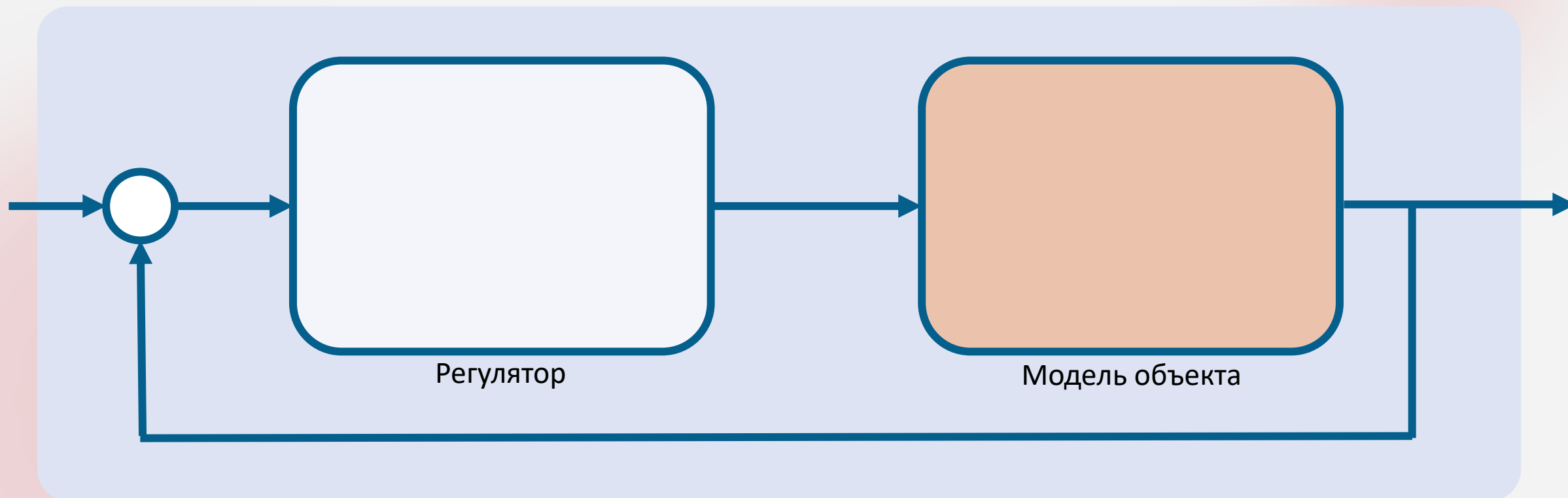


Этапы разработки системы управления

план доклада



- ✓ Построение модели **объекта** управления
- ✓ Разработка, тестирование и верификация **алгоритма** управления



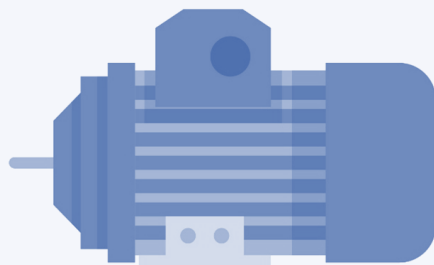
Как моделировать объект?

модель объекта



ЗАДАЧА

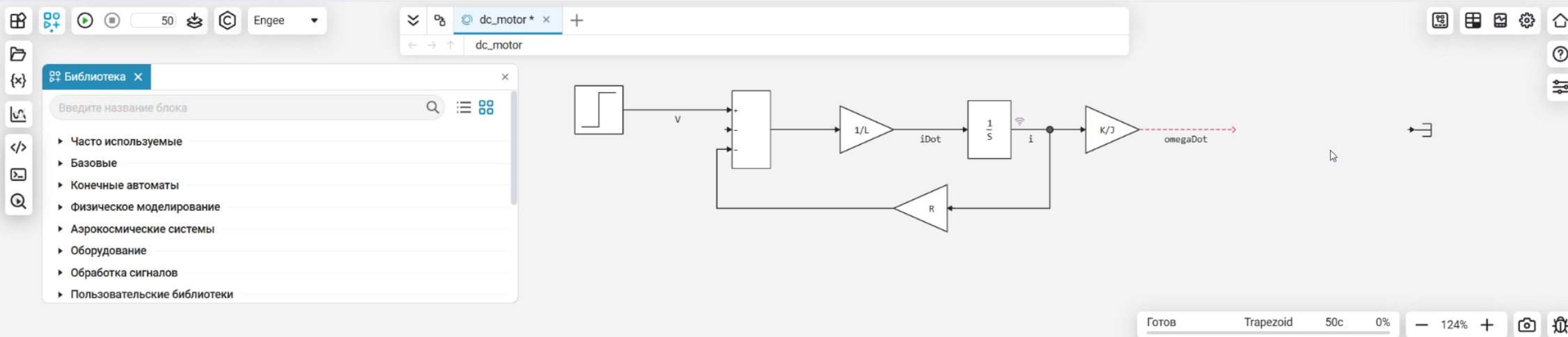
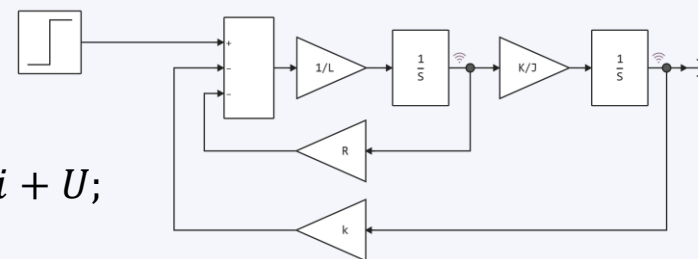
Создать модель
двигателя постоянного
тока



МОДЕЛИРОВАНИЕ МАТЕМАТИЧЕСКИМИ БЛОКАМИ

$$J \frac{d\omega}{dt} = Ki;$$

$$L \frac{di}{dt} = -k\omega - Ri + U;$$



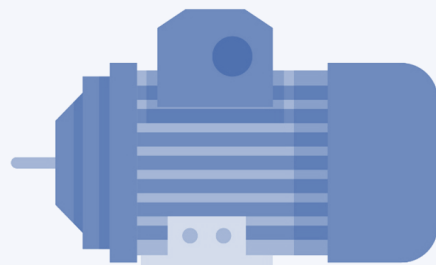
Как моделировать объект?

модель объекта



ЗАДАЧА

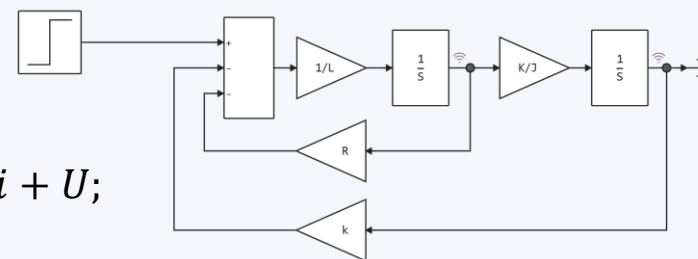
Создать модель
двигателя постоянного
тока



МОДЕЛИРОВАНИЕ МАТЕМАТИЧЕСКИМИ БЛОКАМИ

$$J \frac{d\omega}{dt} = Ki;$$

$$L \frac{di}{dt} = -k\omega - Ri + U;$$



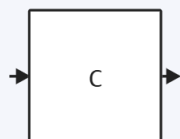
ПЕРЕИСПОЛЬЗОВАНИЕ ГОТОВОГО КОДА



Engage Function



Fortran



C Function

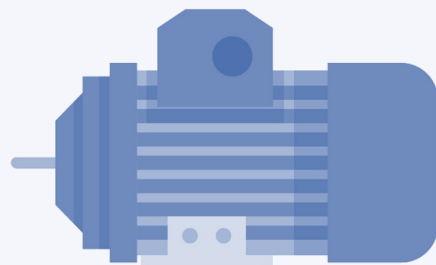
Как моделировать объект?

модель объекта

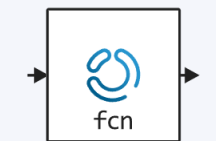
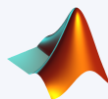


ЗАДАЧА

Создать модель
двигателя постоянного
тока



ПЕРЕИСПОЛЬЗОВАНИЕ ГОТОВОГО КОДА



Engage Function

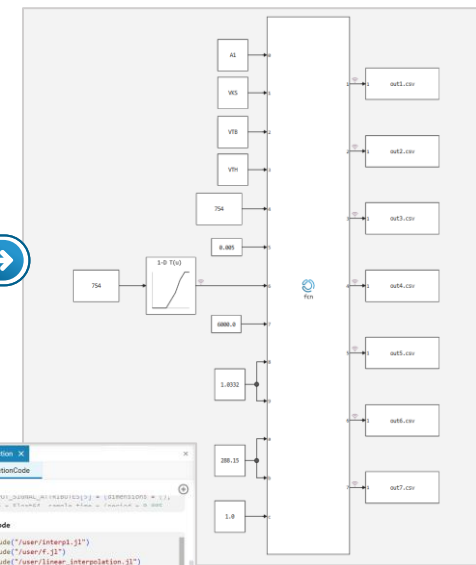
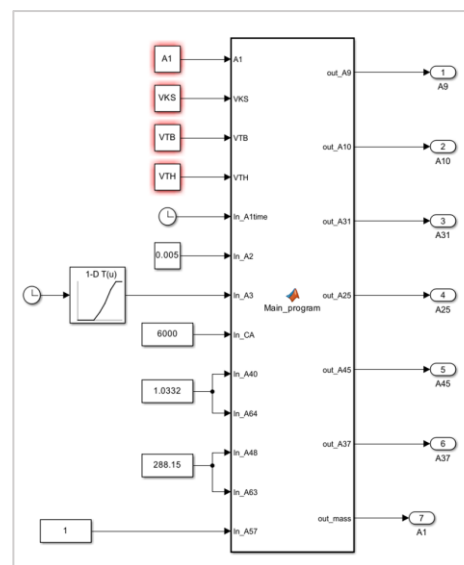


Fortran



C Function

Перенос кода газотурбинного
двигателя из Matlab в Engage



```
EngageFunctionCode
Common code
1 include("user/interp.jl")
2 include("user/f.jl")
3 include("user/linear_interpolation.jl")
4 include("user/interp.jl")
5 include("user/burner.jl")
6 include("user/linear_interp_piece.jl")
7 include("user/integration.jl")
8 include("user/derivatives.jl")
9 include("user/compressor.jl")
10 include("user/turbine.jl")
11 include("user/engine.jl")
12 include("user/input_data.jl")
13
14
15 mutable struct MainProgram <:
16 AbstractCausalComponent
17 A25 :: Float64
18 A45 :: Float64
19 A37 :: Float64
20 B22 :: Float64
21 B57 :: Float64
22 ...
```

Как моделировать объект?

модель объекта



ЗАДАЧА

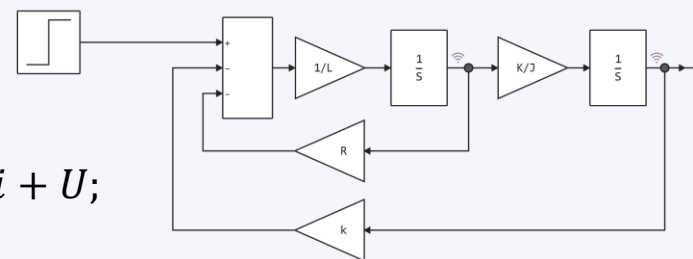
Создать модель
двигателя постоянного
тока



МОДЕЛИРОВАНИЕ МАТЕМАТИЧЕСКИМИ БЛОКАМИ

$$J \frac{d\omega}{dt} = Ki;$$

$$L \frac{di}{dt} = -k\omega - Ri + U;$$



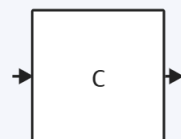
ПЕРЕИСПОЛЬЗОВАНИЕ ГОТОВОГО КОДА



Engage Function



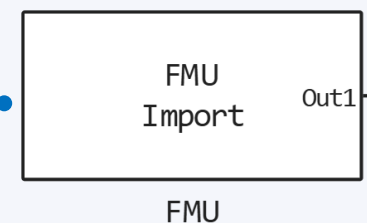
Fortran



C Function



ИМПОРТ МОДЕЛЕЙ ИЗ ДРУГИХ СРЕД



Как быстрее моделировать объект?

модель объекта

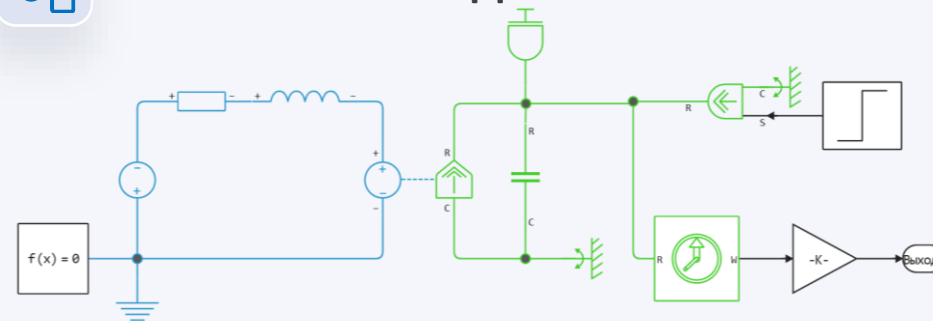


ЗАДАЧА

Создать модель
двигателя постоянного
тока



ФИЗИЧЕСКОЕ МОДЕЛИРОВАНИЕ



Engage

ssc_dcmotor * x

ssc_dcmotor

Библиотека

Введите название блока

- Физическое моделирование
- Аэрокосмические системы
- Навигация
- Оборудование
- Обработка сигналов

Вход1

Выход1

Готов Euler 0.2c 0% 110%

Физическое моделирование

модель объекта



+ Гибкость и адаптивность

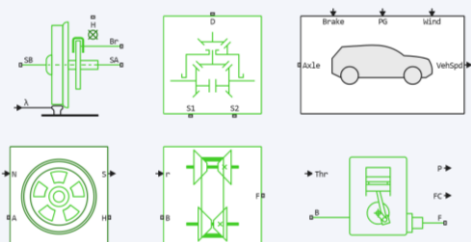
+ Концентрация на сути процессов

+ Быстрое создание и проверка моделей

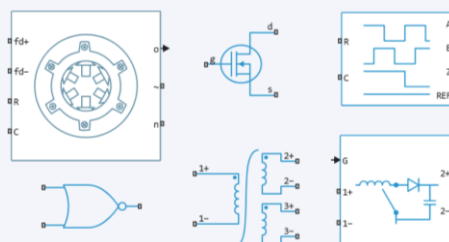
+ Готовые решения для разных областей

+ Высокая скорость и простота моделирования

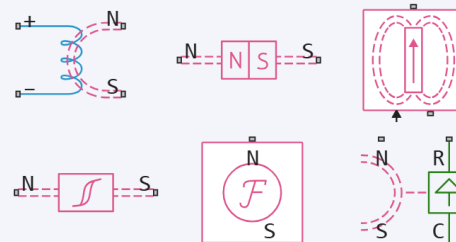
1 Механика



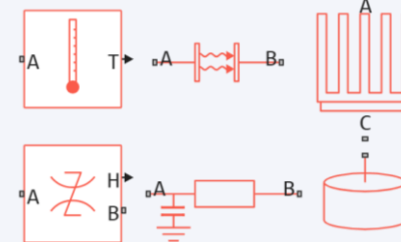
2 Электричество



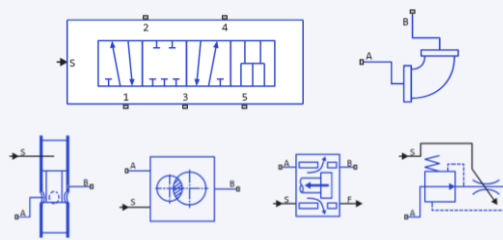
3 Магнетизм



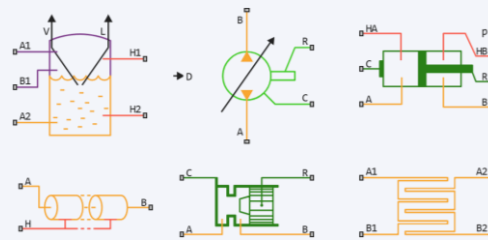
4 Теплотехника



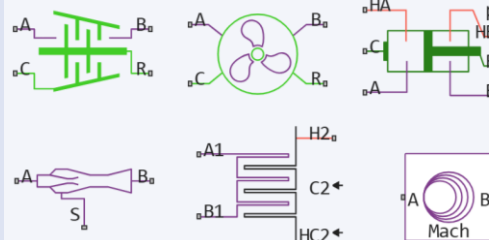
5 Гидравлика



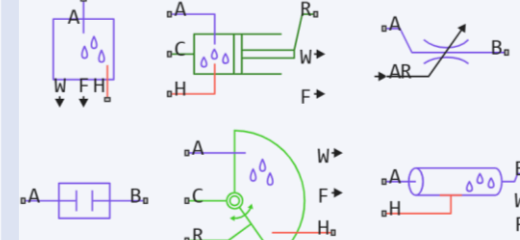
6 Теплогидравлика



7 Газ

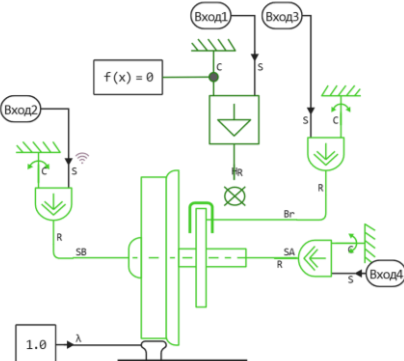
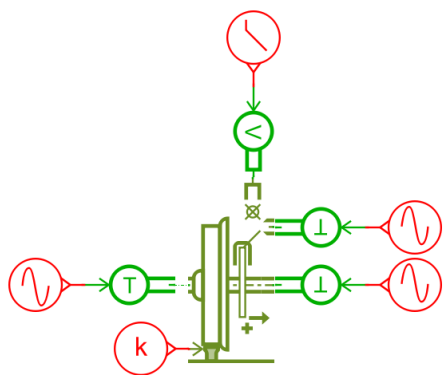


8 Влажный воздух



Физическое моделирование

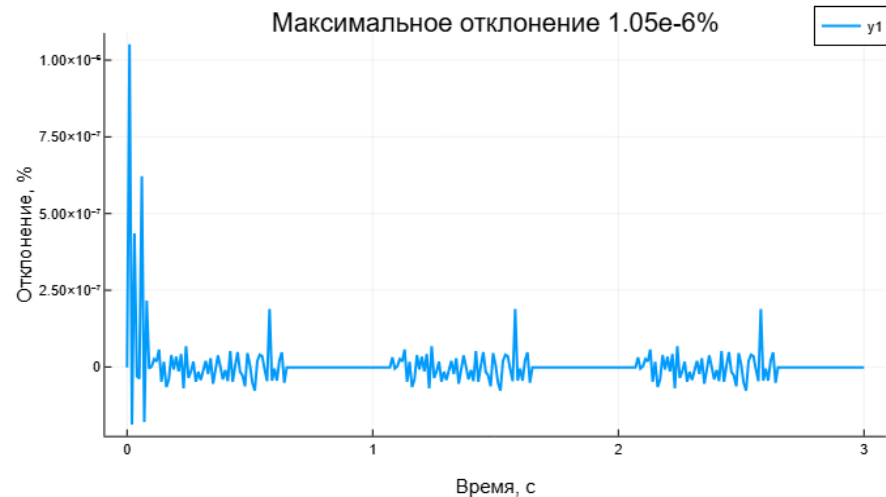
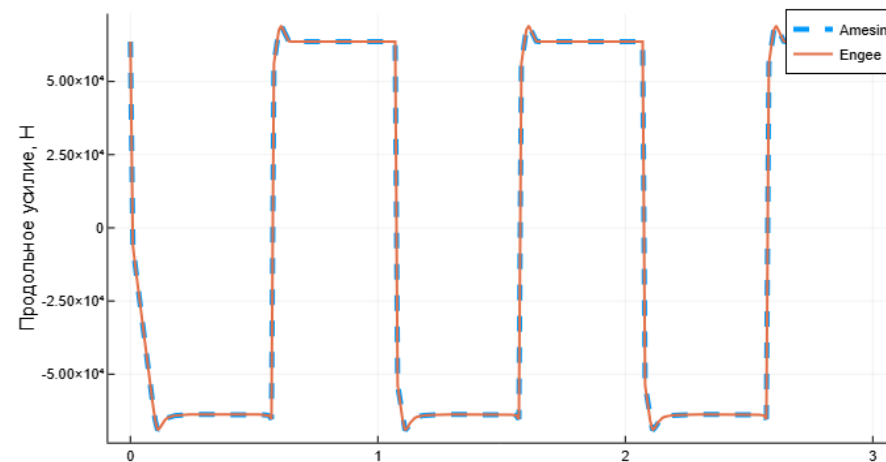
модель объекта



MathWorks®
Simscape

SC
AME
SIEMENS


Engee



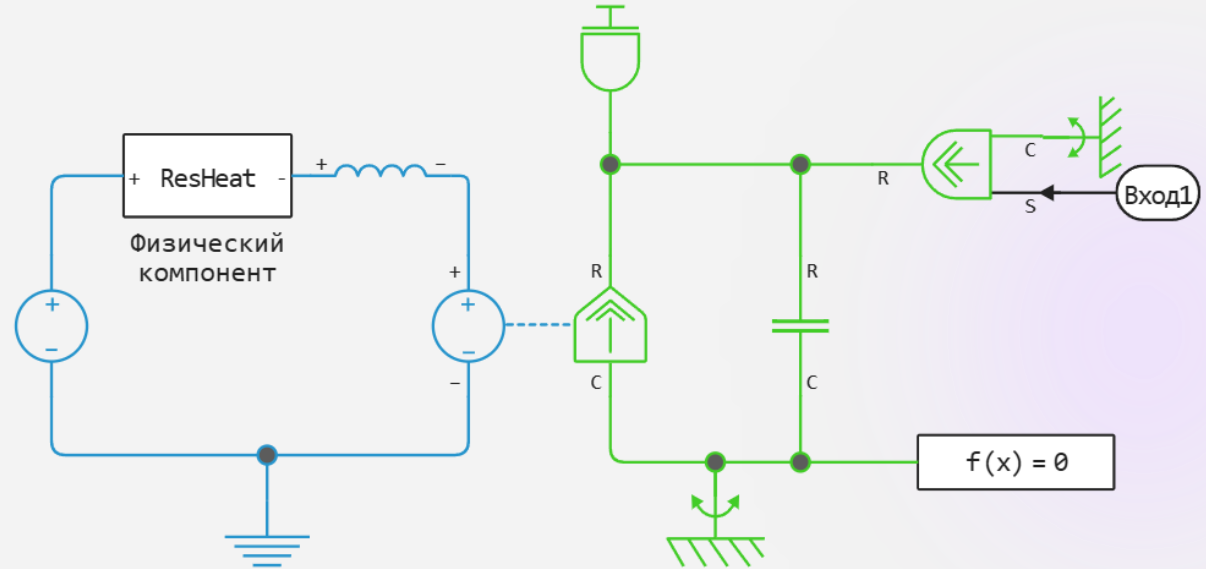
Язык физического моделирования Engage

модель объекта



```
1 @engeemodel ResHeat begin
8   @variables begin
9     v = 0, [unit="V",description = "Voltage"]
10    i = 0, [unit="A",description = "Current"]
11  end
13  @parameters begin
14    R = 200.0, [unit::String="Ohm"]
15    a = 0.1, [unit::String="1/K"]
16    T0 = 300, [unit::String="K"]
17    T = 300, [unit::String="K"]
18  end
24  @equations begin
25    v ~ p.v-n.v
26    v ~ R*(1+a*(T-T0))*i
27  end
```

$$V(t) = R \cdot (1 + \alpha \cdot (t - t_0)) \cdot i(t)$$



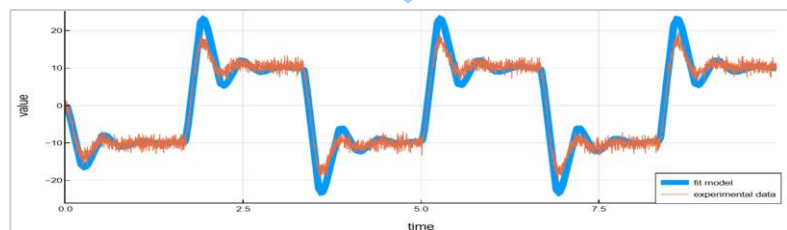
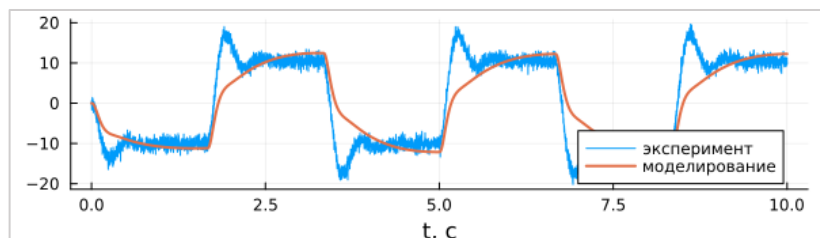
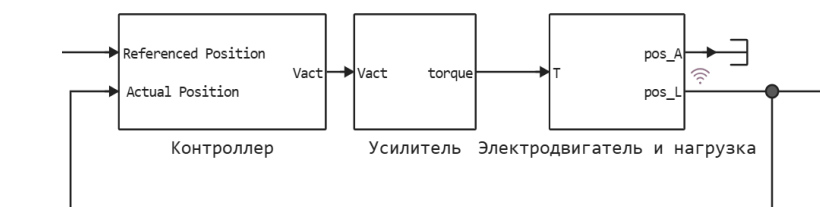
- ✓ Может описать уникальную математическую модель
- ✓ Декларативный: описывает связи между величинами, а не алгоритм
- ✓ Вдохновлён Modelica и Simscape

Инструменты отладки физических моделей

модель объекта



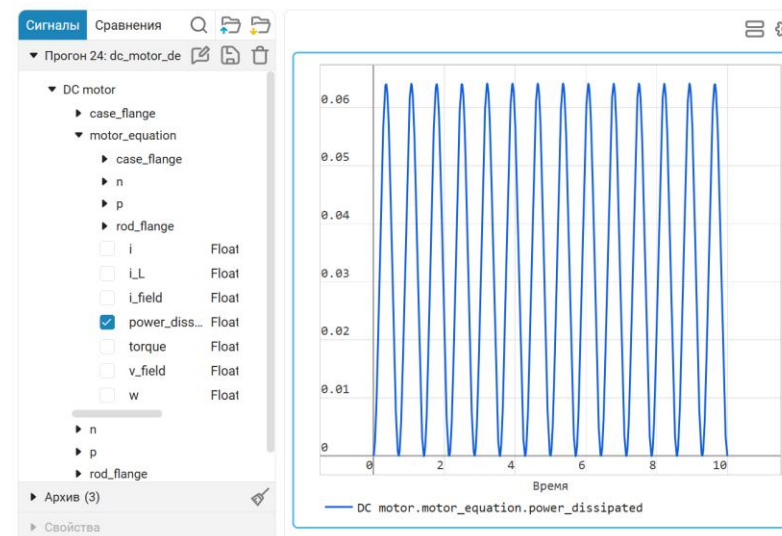
- Инструменты подбора параметров модели по экспериментальным данным.



Имя	Статус	Приоритет	Целев...	Нача...	Исключена
Инерция → flange → w	✓	None	0	0.0069...	✓
Инерция → T	✓	None	0	-0.000...	✓
Инерция → w	✗	High	0.01047...	0.0069...	✓
AC Voltage Source → i	✓	None	0	4.7999...	✓
AC Voltage Source → n → v	✓	None	0	0	✓
AC Voltage Source → p → v	✓	None	0	0	✓
AC Voltage Source → v	✓	None	0	0	✓
DC motor → case_flange → w	✓	None	0	0	✓
DC motor → motor_equation → case...	✓	None	0	0	✓
DC motor → motor_equation → i	✓	None	0	-4.799...	✓

- Таблица физических переменных
- Решение конфликтов инициализации

- Запись внутренних переменных
- Инспектор данных



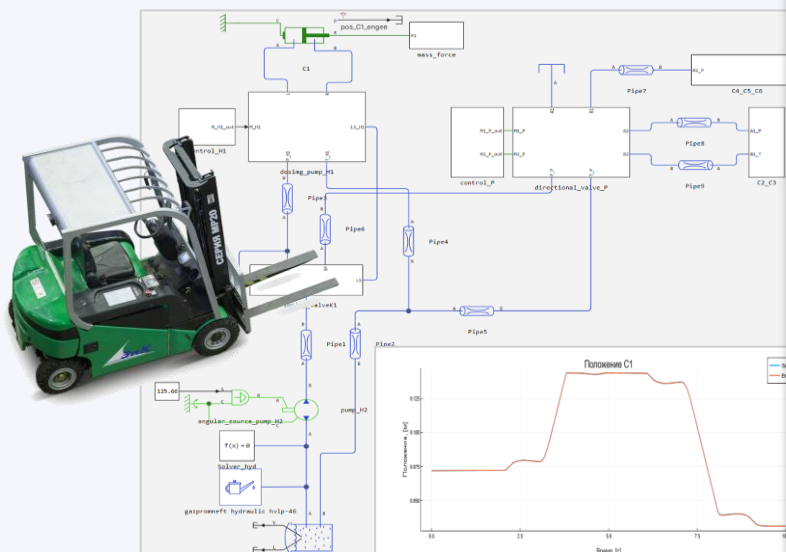
Истории успеха

модель объекта



Модель гидропривода электропогрузчика

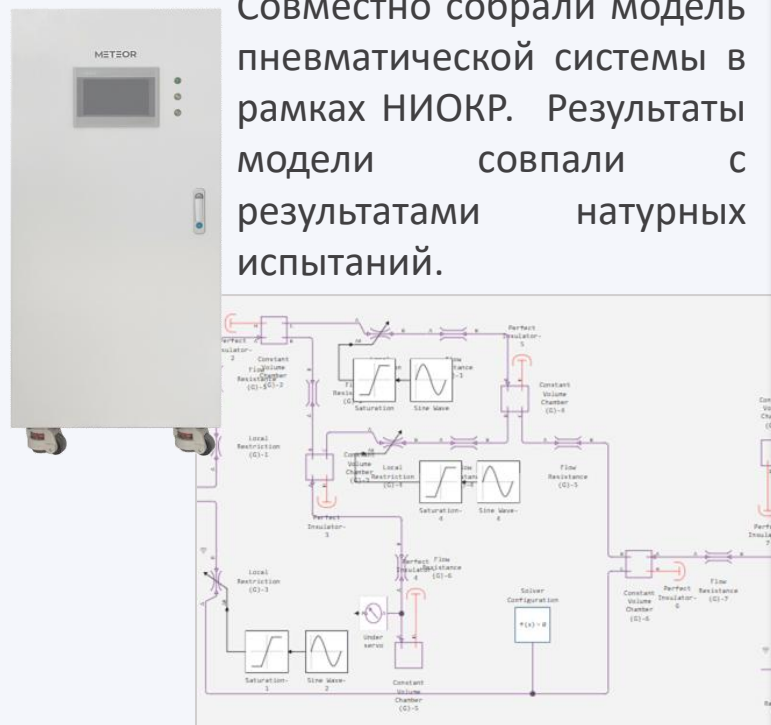
Собрали модели гидропривода в Engage и MATLAB/Simscap. Доказали, что результаты симуляции идентичные.



Газовый регулятор котла отопления

METEOR THERMO

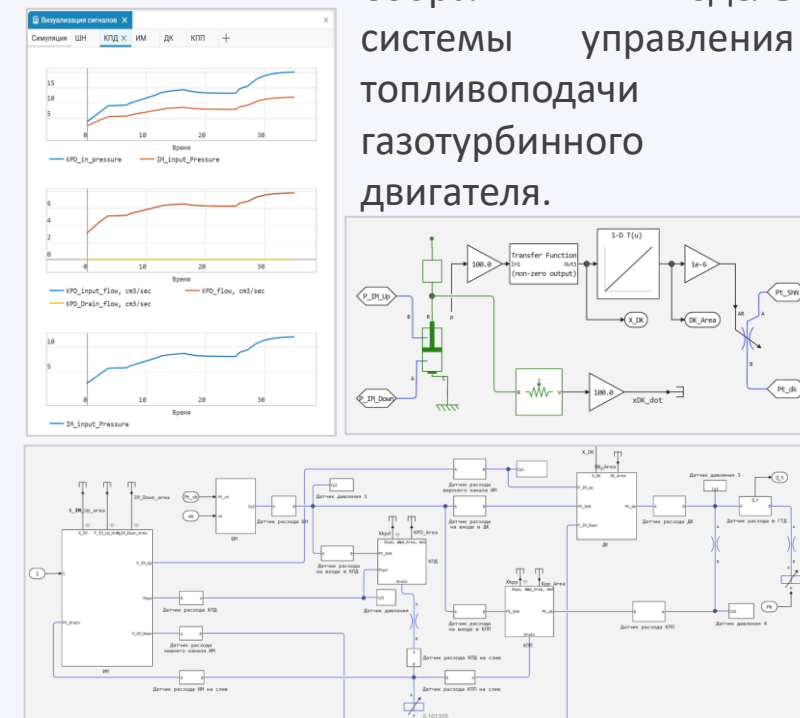
Совместно собрали модель пневматической системы в рамках НИОКР. Результаты модели совпали с результатами натурных испытаний.



Модель гидропривода



Собрали модель системы управления топливоподачи газотурбинного двигателя.

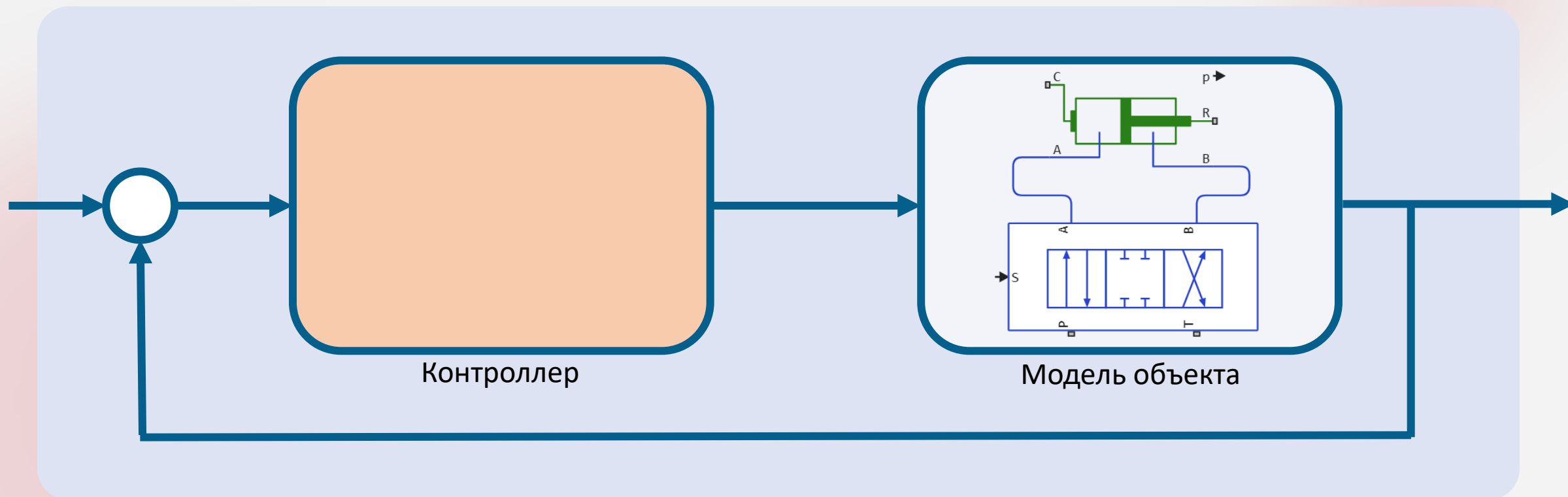


Этапы разработки системы управления

план доклада



- ✓ Построение модели объекта управления
- ✓ Разработка, тестирование и верификация алгоритма управления



Прикладные библиотеки



- + Системы управления: EngEEControlSystems.jl, ControlSystems.jl
- + Чтение/запись данных: DataFrames.jl, CSV.jl
- + Визуализация данных: Plots.jl, Makie.jl
- + Линейная алгебра: LinearAlgebra.jl
- + Оптимизация: Optim.jl, JuMP.jl
- + Графы: Graphs.jl
- + Дифференциальные уравнения: DifferentialEquations.jl
- + Символьно-численное моделирование: ModelingToolkit.jl
- + Метод конечных элементов: JuliaFEM.jl
- + Нейронные сети: Flux.jl
- + Химические реакции: Catalyst.jl.

The screenshot shows the Julia REPL interface with a script editor on the right. The script, named `margin_of_stability.ngscript`, contains the following code:

```
1 Pkg.add(["ControlSystems"])
2 using ControlSystems

1 G = tf([.5, 1.3], [1, 1.2, 1.6, 0]);
2 T = feedback(G, 1)

Получим полюса системы.

1 pole(T)

1 plot(rlocus(G))

1 marginplot(G)

1 S1 = stepinfo(step(T, 25));
2 plot(S1)

1 M = margin(2*G)
2 GmdB = 20*log(10, M.gm[1][1])
3 Pm = M.pm[1][1]
4 display(GmdB)
5 display(Pm)
```

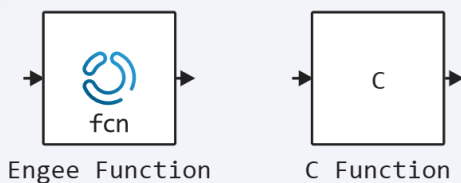
The interface includes a left sidebar with tabs for "Библиотека" (Library), "Переменные" (Variables), and "Рабочая область" (Workspace). The right sidebar shows the script editor with a toolbar and execution controls. The status bar at the bottom indicates "Код" (Code), "Выполнено" (Executed), "Последняя выполненная ячейка" (Last executed cell), and "Активная ячейка" (Active cell).

Моделирование алгоритмов управления

управление

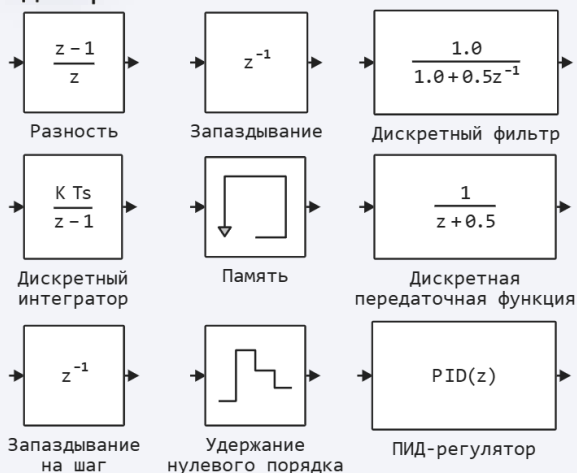


ПОЛЬЗОВАТЕЛЬСКИЕ ФУНКЦИИ

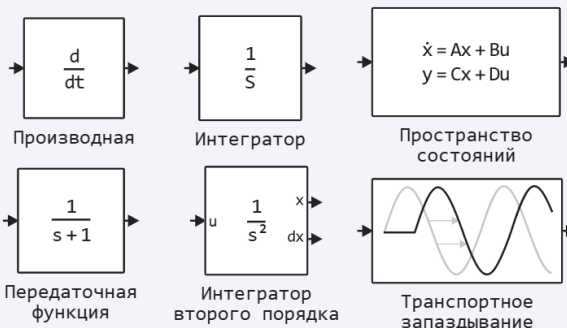


ГИБРИДНЫЕ СИСТЕМЫ

Дискретные

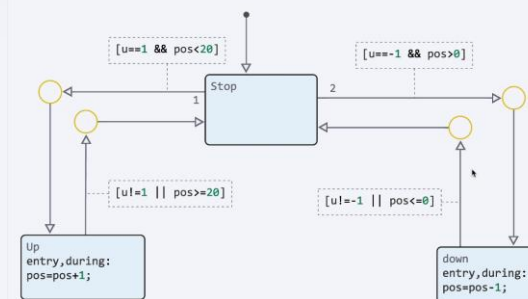


Непрерывные



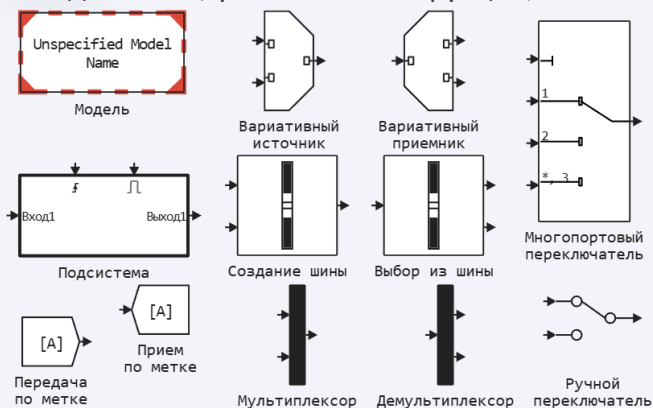
КОНЕЧНЫЕ АВТОМАТЫ

Обработка отказов, управляющая логика



АРХИТЕКТУРА И МАРШРУТИЗАЦИЯ

Подсистемы, условные конструкции, шины



ДОПОЛНИТЕЛЬНО

Атрибуты сигналов

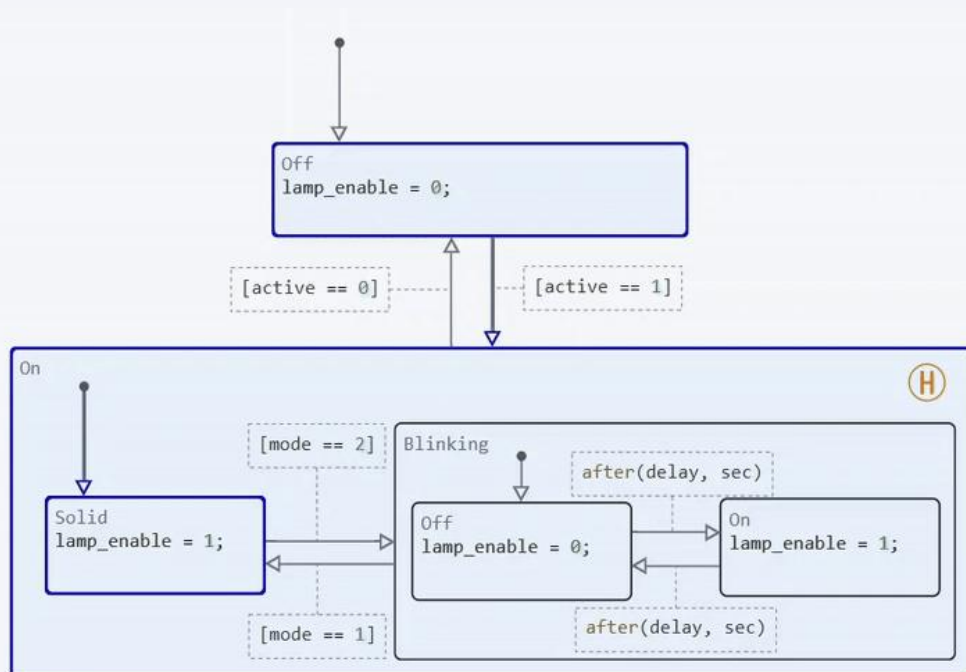


Таблицы

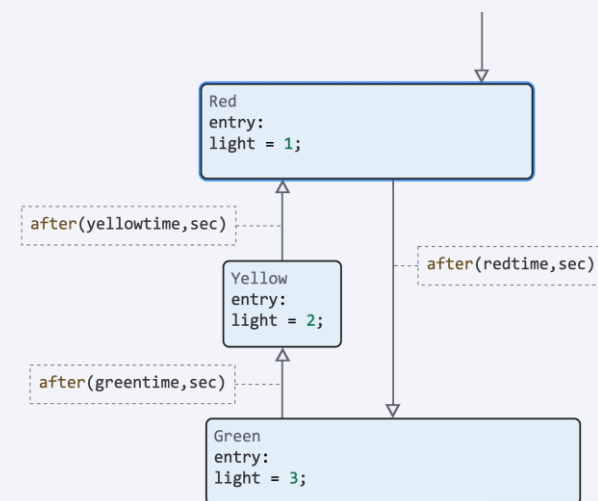




КОНЕЧНЫЕ АВТОМАТЫ



- ✓ Управляющая логика
- ✓ Обработка отказов
- ✓ Режимы работы систем



Инструменты для разработки САУ

управление



Идентификация

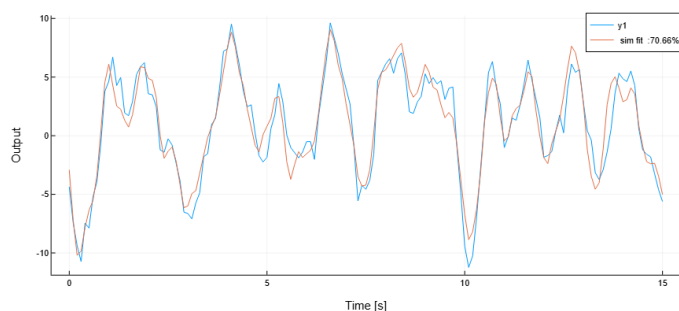
Инструменты построения моделей по экспериментальным данным.

Сбор данных эксперимента

Выбор структуры модели

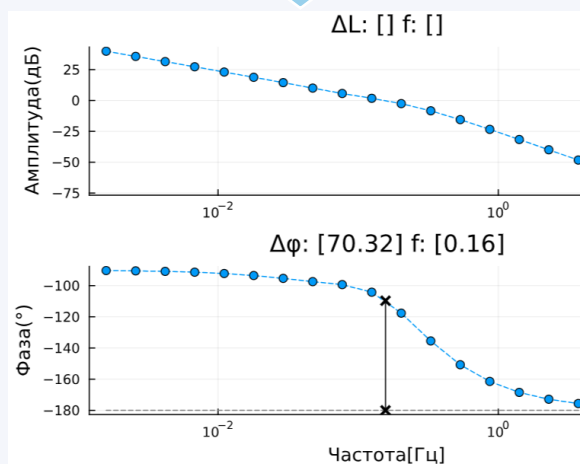
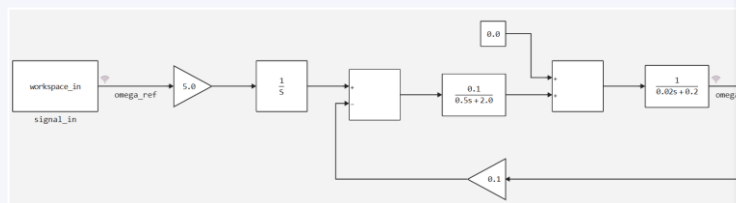
Идентификация модели

Валидация модели



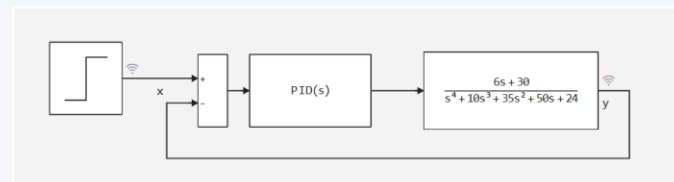
Построение частотных характеристик

Функции для получения частотных характеристик системы.



Командное управление

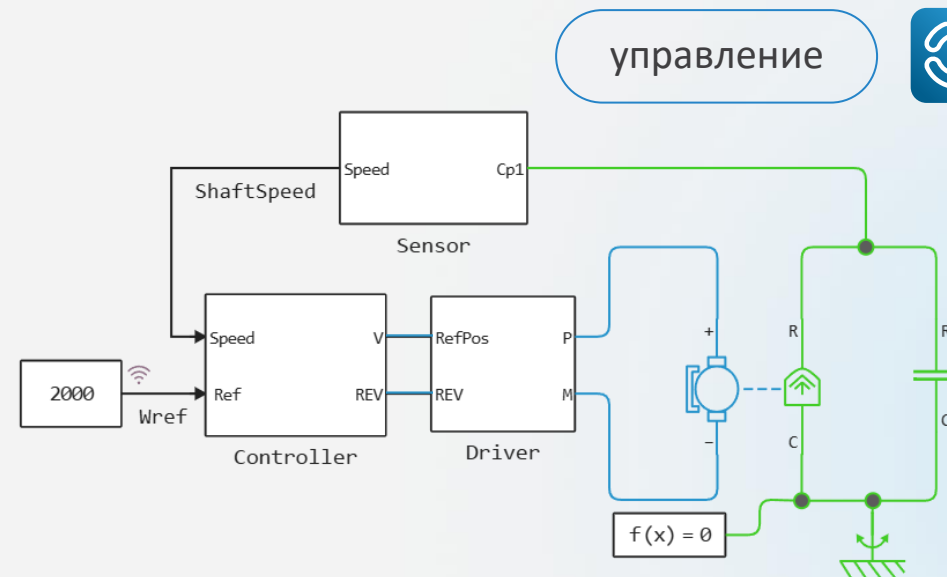
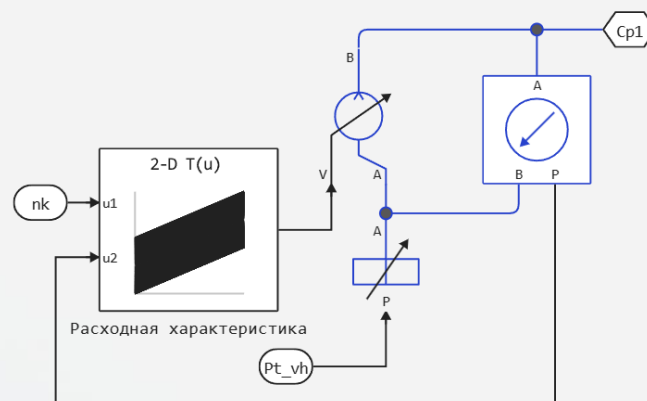
Возможности автоматизации настройки, запуска моделей и получения результатов.



Все в одной модели!



Мультидоменное
моделирование
алгоритмов
и физических систем



НЕПРЕРЫВНЫЕ МОДЕЛИ

Системная динамика
Модели окружения



ДИСКРЕТНЫЕ МОДЕЛИ

Цифровое управление
ЦОСР
РЛС
Компьютерное зрение



ФИЗИЧЕСКИЕ МОДЕЛИ

Электроника
Механика
Гидравлика
Электрические машины



КОНЕЧНЫЕ АВТОМАТЫ

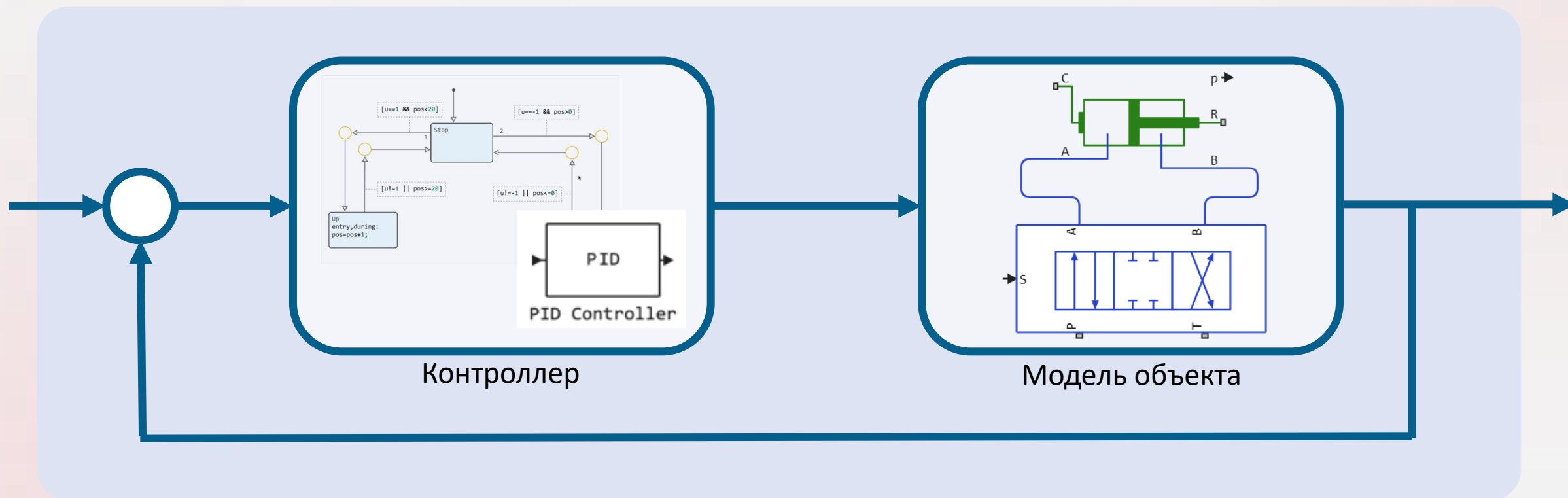
Управляющая логика
Обработка отказов
Режимы работы систем

Что дальше?

тестирование



- ✓ Построение модели объекта управления
- ✓ Разработка, тестирование и верификация алгоритма управления

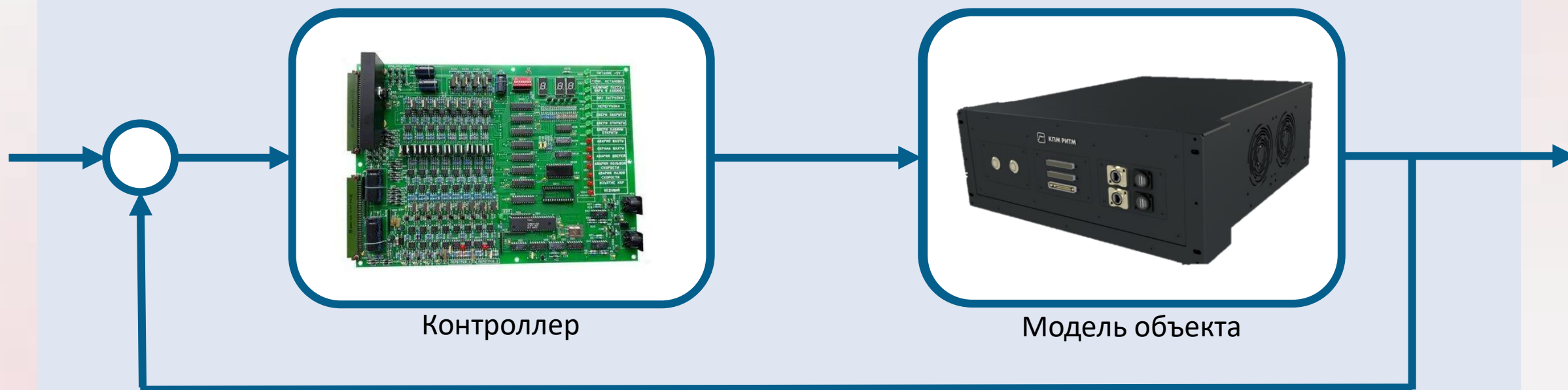


HiL – стенд

тестирование



- ✓ Построение модели объекта управления
- ✓ Разработка, тестирование и верификация алгоритма управления
- ✓ Генерация С-кода
- ✓ Тестирование алгоритмов без физического прототипа



HiL - стенд

тестирование



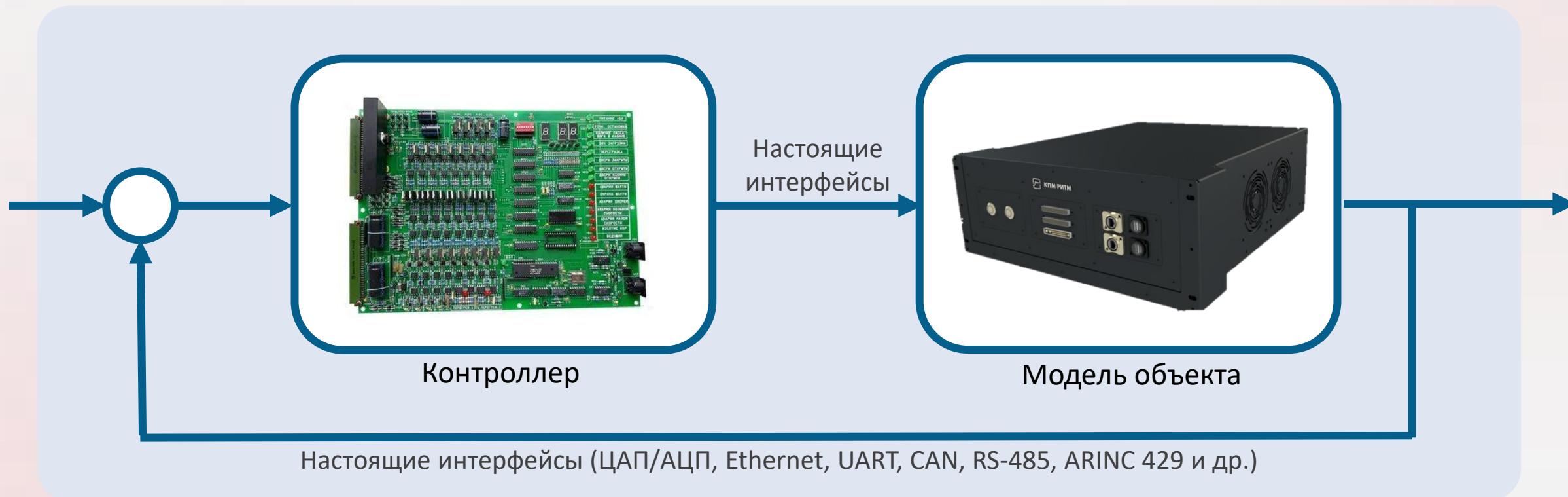
Алгоритм протестирован в реалистичных условиях с аппаратной интеграцией, в отказных режимах

Учтены:

- ✓ Цифровые эффекты (квантование, задержки АЦП/ЦАП)
- ✓ Шумы датчиков и помехи в цепях
- ✓ Производительности ЦПУ

Обнаружены ошибки:

- ✓ Временные
- ✓ Взаимодействие с драйверами
- ✓ Переполнение...

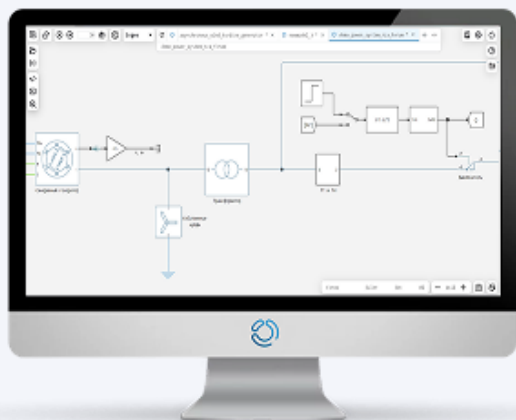


Программно-аппаратный комплекс полунатурного моделирования РИТМ

тестирование



- 🛡️ Снижаем риски
- 💰 Экономим время



Engage

Предметные цифровые модели
систем и алгоритмов



КПМ РИТМ

Имитационные, управляющие
и испытательные стенды для отладки
технических систем на базе КПМ РИТМ



Стенды в авиации
<https://new.exponenta.ru/aviation>

Стенды в автомобильной сфере
<https://new.exponenta.ru/automobile>



ЭКСПОНЕНТА

ЦЕНТР ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ
И МОДЕЛИРОВАНИЯ



Спасибо за внимание!

