

Ну, заяц, погоди!

ИЛИ

Создаем цифровые двойники энергообъектов

Валерий Очков, Дарья Абрамова
НИУ «МЭИ»
Вячеслав Петухов
ООО «ЗВ Сервис»

Аннотация. В статье рассказано, как с помощью двух отечественных компьютерных программ можно решать интересные математические задачи и создавать тренажёры для персонала энергетических предприятий.

«— Ату его, — слышался в это время протяжный крик одного из остановившихся борзятников. Он стоял на полубугре жнивья, подняв аранник, и ещё раз повторил протяжно: — А-ту — его!»

Лев Толстой «Война и мир»

Рассмотрим задачу, которая на первый взгляд кажется простой, но на самом деле требует серьезных усилий для полного решения.

На газоне с помощью вращающейся консоли по кругу волочат с постоянной скоростью тряпичного зайца (рис. 1). В какой-то момент из любого места площадки – из её центра, например, выпускают собаку (настоящую или четырёхлапого кибер-робота) и направляют её на зайца – см. эпиграф. Как побежит собака, если вектор её движения направлен строго на зайца, а скорость тоже постоянна и может быть выше, ниже или равна скорости зайца.



Рисунок 1 – Погоня собак за тряпичным зайцем

Это хорошо известная, так называемая *задача о погоне*, простейший случай которой такой – заяц бежит по прямой, а сбоку на него выскакивает собака. В такой ситуации решение сводится к составлению несложного обыкновенного дифференциального уравнения второго порядка с начальными условиями, которое имеет аналитическое решение¹. Однако, усложнение траектории движения цели (например, движение по окружности) приводит к необходимости применения численных методов решения дифференциальных уравнений, поскольку аналитическое решение в таких случаях, как правило, отсутствует. Одним из простейших численных методов, применимых для данной задачи, является метод Эйлера, который он придумал, живя и работая в Санкт-Петербурге (1768 г.). Но можно убрать этап составления дифференциального уравнения и сразу решать задачу о погоне с помощью разностной схемы² с использованием приёмов дискретной математики. Что мы и сделаем!

Для решения поставленной задачи будут использованы два программных средства: в качестве первого способа будет использована система компьютерной математики SMath Studio, являющаяся российским аналогом Mathcad, а в качестве второго способа – среда визуального программирования SimInTech, аналогичная Simulink. И SMath Studio, и SimInTech созданы задолго до эры санкций. Санкции только придали им второе дыхание по принципу «нет худа без добра»³. SMath Studio распространяется бесплатно через сайт www.smath.com, а SimInTech доступен для скачивания в пробной версии на сайте www.simintech.ru.

Способ 1. На рисунке 2 показано решение задачи о погоне в среде SMath Studio. Для этого генерируются точки траекторий движения зайца (*hare*) и бегущей за ней собаки (*dog*). Затем эти траектории отображаются на графиках и в анимации (рис. 3–7).

На первой строке расчёта (рис. 2) задается радиус окружности, по которой бежит заяц – 700 метров. Единица длины метр (синяя, а не чёрная буква *m*) вводится в расчёт через кнопку ввода единиц измерения с изображением воронки (см. правую иконку на верхней панели программы SMath Studio или сочетание клавиш CTRL+W). Далее задаются скорости зайца и собаки (переменные с текстовыми индексами, которые вводятся через точку) с

¹ <https://community.ptc.com/t5/Mathcad/Wolf-and-hare-one-old-problem-with-simple-Mathcad-solution/m-p/574235>

² В. Ф. Очков, И. Е. Васильева. Применение разностных схем к решению задачи о погоне // Труды СПИИРАН. Выпуск 18 (6), 2019. С. 1406-1429 (<https://ia.spcras.ru/index.php/sp/article/view/4041>)

³ Примечательный факт. Три «кита» САПР – программы AutoCAD, SolidWorks и Creo (бывшая Pro/Engineer) зарождались в СССР, но потом переместились на Запад.

единицами скорости. Числа, естественно, условные. Единицы измерения помогут избежать ошибок типа сложения метров с килограммами. Переменная Δt – это заданная длительность скачка собаки и зайца во время бега. А делают они n скачков за один сеанс гонки. Для повышения точности расчёта можно уменьшать значение Δt и одновременно увеличивать значение n .

$R := 700 \text{ m}$ $v_{hare} := 5 \frac{\text{m}}{\text{s}}$ $v_{dog} := 5.35 \frac{\text{m}}{\text{s}}$ $\Delta t := 1 \text{ s}$ $n := .50$ (1)

$\Delta\alpha := \frac{v_{hare} \cdot \Delta t}{R} = 0.4093^\circ$ **v.hare** $i := [1..n]$ (2)

v.hare[1] $x_{hare}_i := R \cdot \sin(\Delta\alpha \cdot i)$ $y_{hare}_i := R \cdot \cos(\Delta\alpha \cdot i)$ $x_{dog}_1 := 0$ $y_{dog}_1 := 0$ (3)

$\varphi_1 := \left(\text{arctg} \left(\frac{y_{hare}_1 - y_{dog}_1}{m}, \frac{x_{hare}_1 - x_{dog}_1}{m} \right) \right)$ (4)

Программирование
 for $i \in [1..n]$ if for try

$\begin{bmatrix} x_{dog}_{i+1} \\ y_{dog}_{i+1} \\ \varphi_{i+1} \end{bmatrix} := \begin{bmatrix} x_{dog}_i + v_{dog} \cdot \Delta t \cdot \cos(\varphi_i) \\ y_{dog}_i + v_{dog} \cdot \Delta t \cdot \sin(\varphi_i) \\ \text{arctg} \left(\frac{y_{hare}_i - y_{dog}_i}{m}, \frac{x_{hare}_i - x_{dog}_i}{m} \right) \end{bmatrix}$ **Матрицы** (5)

$XY_{hare} := \text{augment}(x_{hare}, y_{hare}, ".", 5, \text{"gray"})$ (6)

$XY_{hareN} := \text{augment}(x_{hare}_n, y_{hare}_n, ".", 15, \text{"gray"})$ (7)

$XY_{dog} := \text{augment}(x_{dog}, y_{dog}, ".", 5, \text{"red"})$ (8)

$XY_{dogN} := \text{augment}(x_{dog}_n, y_{dog}_n, ".", 15, \text{"red"})$ (9)

$S := \sqrt{(x_{hare}_n - x_{dog}_n)^2 + (y_{hare}_n - y_{dog}_n)^2} = 445.0844 \text{ m}$ (10)

$P := \begin{bmatrix} x_{hare}_n & y_{hare}_n \\ x_{dog}_n & y_{dog}_n \end{bmatrix}$ (11)

$plot := \begin{cases} XY_{hare} & (12) \\ XY_{hareN} \\ XY_{dog} \\ XY_{dogN} \\ P \end{cases}$

Функции
 log sign
 ln arg
 exp {

Вставка третьего, четвертого и т.д. элемента через нажатие клавиши "запятая"!

Рисунок 2 – Расчёт траектории погони собаки за зайцем.

На второй строке расчёта определяется значение переменной $\Delta\alpha$ – угла разворота зайца относительно центра круга при каждом скачке (отношения пройденного пути $v_{hare}\Delta t$ к длине окружности R). Затем переменной i (номер очередного скачка зайца и собаки) присваиваются значения 1, 2, 3 и т.д. до n . Это делается с помощью кнопки ввода диапазона значений с единичным шагом в панели инструментов **Матрицы**. Фрагмент этой и других панелей инструментов показан на рисунке 2.

На третьей строке расчёта заполняются векторы x_{hare} и v_{hare} (координаты зайца при каждом его скачке) и первые элементы векторов x_{dog} и v_{dog} – заданные начальные координаты собаки (собака, напомним, стартует из центра круга, который является центром координат). Но можно менять стартовые координаты собаки и строить новые траектории её бега за зайцем. Несложно также изменить значения векторов x_{hare} и v_{hare} так, чтобы заяц бежал не по кругу, а по квадрату или по другой иной замкнутой или разомкнутой кривой, что будет показано ниже. Оператор обращения к элементу вектора – это также индекс, но вводится он в расчёт, не точкой (см. выше), а через клавишу открывающейся квадратной скобки или через нажатие определенной клавиши панели инструментов Матрицы. В расчёте можно видеть двойные индексы, первый из которых текстовый (часть имени переменной, сдвинутый вниз), а второй – это оператор работы с элементом вектора.

Вектор φ (см. пункт 4) будет хранить угол направления бега собаки к зайцу относительно горизонтальной линии (см. рис. 3 справа вверху). Такой угол называется *азимутом* – углом между заранее выбранным направлением (у нас это восточное направление – см. рис. 3) и направлением на заданный объект. Начальное значение этого угла и определяется оператором на четвертой строке расчёта. Используется функция арктангенса, но не с одним, а с двумя аргументами. Функция арктангенса с двумя аргументами не работает с размерными величинами (недоработка программы SMath Studio). Поэтому эти аргументы через деление на метр лишаются размерности.

Пятый оператор расчёта («гвоздь программы») рекуррентно формирует три вектора x_{dog}, v_{dog} и φ , первые элементы которых заданы ранее. Эти векторы нельзя заполнить отдельными операторами так, как это сделано для векторов x_{hare} и v_{hare} (см. строку 3). Будет появляться сообщение об ошибке. Векторы x_{dog}, v_{dog} и φ нужно заполнять одновременно, меняя при этом значение переменной i . Поэтому операторы для их заполнения вставлены в тело цикла с параметром *for* и объединены в два вектора с тремя элементами. Зная положение собаки перед очередным скачком в точке i и азимут в этой

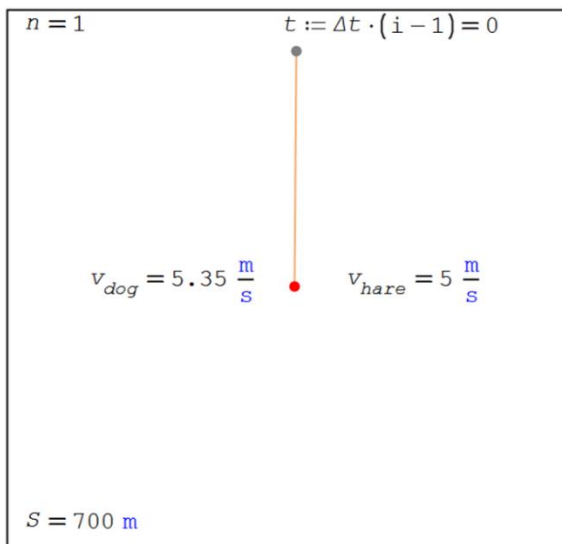
точке (угол φ), можно рассчитать положение собаки после очередного скачка движения в точке $i + 1$. Вот и весь алгоритм расчета!

Операторы 6-11 предназначены для построения графиков по рассчитанным данным – траекторий бега зайца и собаки (операторы 6 и 8), изображения кружочками размером 15 единиц самого зайца и собаки (операторы 7 и 9). Заяц рисуется серым цветом (gray), а собака красным (red). Встроенная в SMath Studio функция *augment* соединяет несколько векторов или скаляров в единую матрицу, по которой строятся графики, отдельные точки которой соединяются прямыми линиями (умолчание) или отрезками кубического полинома (выбор пользователя).

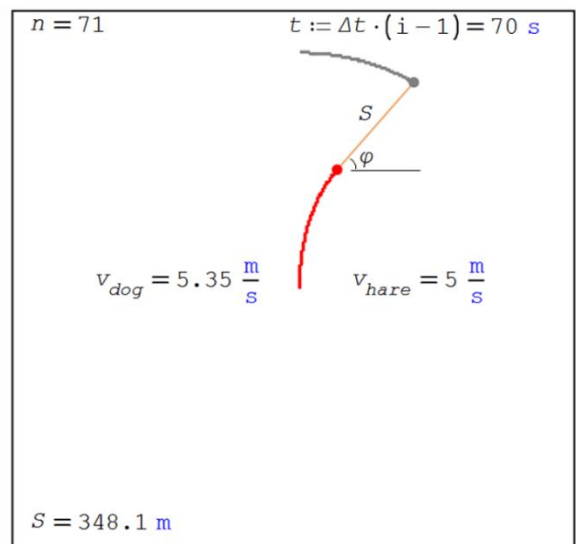
Переменная *S* (пункт 10) хранит расстояние от собаки до зайца, которое пунктиром будет отображаться на графике вместе с другими переменными. Соответствующие операторы нужно создать отдельно, а потом перетащить на график. Матрица *P* (пункт 11) хранит текущие координаты зайца и собаки, что позволяет построить прямую линию, соединяющую собаку с зайцем (линию азимута). Переменная *plot* объединяет сформированные ранее переменные для построения графика бега собаки за зайцем.

Графики на рисунках 3 и 4 – это штатные графики пакета SMath, которые вставляются в расчёт командами меню Вставить / График / Двухмерный (2D). После вставки в расчёт графика и его аргумента Plot кривых не будет видно, так как начальная область графика ограничена значениями от -10 до 10 по горизонтали и вертикали. Чтобы изменить область графика, необходимо активировать график щелчком мыши по нему и поработать колесиком мышки (zoom). При этом если зажимать клавиши Shift или Ctrl, то это будет приводить к изменению области охвата по горизонтали или по вертикали. На рис. 3 показаны четыре кадра анимации погони собаки за зайцем. На 518 секунде собака поймаёт зайца, когда расстояние между ними сократится до 4 мм.

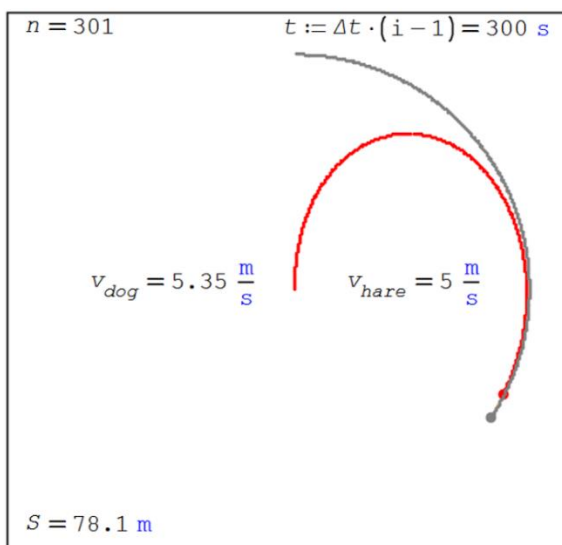
Если скорость собаки будет меньше скорости зайца, то собака, сделав некий крюк, выйдет также на стабильное круговое движение (рис. 4). Радиус окружности, по которой будет бегать собака сзади и сбоку от зайца (пеленг), определить несложно, учитывая тот факт, что у зайца и собаки при разных линейных скоростях будут одинаковые угловые скорости.



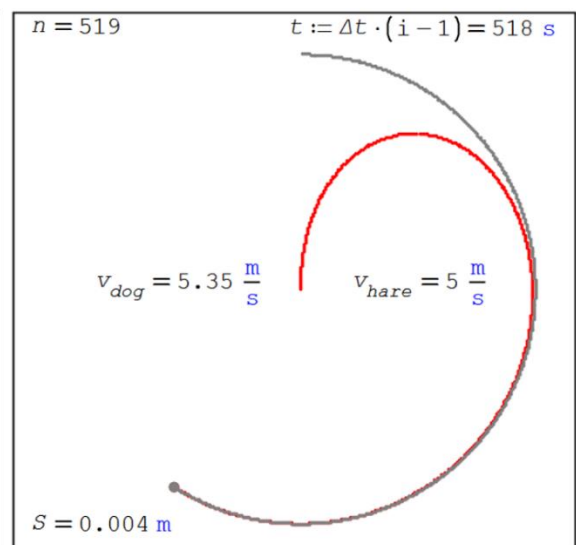
Plot



Plot



Plot



Plot

Рисунок 3 – Кадры анимации погони собаки за зайцем

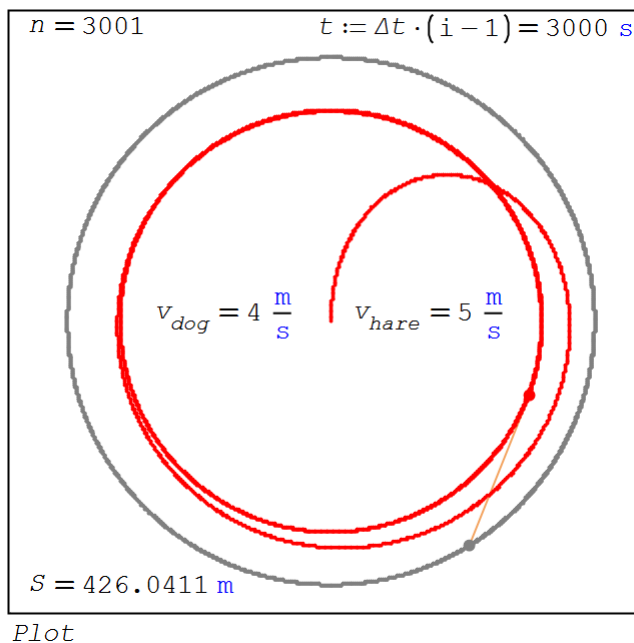


Рисунок 4 – Кадр анимации задачи: собака выбегает на круг

Погоню собаки за зайцем лучше наблюдать в анимации. На рисунке 5 показан SMath-документ, позволяющий её построить. Для этого в нём после ввода исходных данных (без ввода значения n) и расчёта значений трёх переменных, не зависящих от значения n , создается функция пользователя с именем *frame* (кадр анимации) и с аргументом n . Остальные операторы такие же, как в расчёте на рисунке 2.

```

R := 700 M      v_hare := 5  $\frac{M}{C}$       v_dog := 5  $\frac{M}{C}$       Δt := 1 c

Δα :=  $\frac{v_{hare} \cdot \Delta t}{R}$       x_dog_1 := -R      Y_dog_1 := 0 M

frame (n) :=
:= [ i := [1..n]
    [ x_hare_i := R · sin(Δα · i)      Y_hare_i := R · cos(Δα · i) ]
    for i ∈ [1..n]
      [ x_dog_{i+1}      Y_dog_{i+1}      φ_{i+1} ] := [
        x_dog_i + v_dog · Δt · cos(φ_i)
        Y_dog_i + v_dog · Δt · sin(φ_i)
        arctg(  $\frac{Y_{hare}_i - Y_{dog}_i}{M}$ ,  $\frac{x_{hare}_i - x_{dog}_i}{M}$  ) ]
      { augment (x_hare, Y_hare, ".", 5, "gray")
        augment (x_hare_n, Y_hare_n, ".", 15, "gray")
        augment (x_dog, Y_dog, ".", 5, "red")
        augment (x_dog_n, Y_dog_n, ".", 15, "red")
      }
      [ [ x_hare_n      Y_hare_n ]
        [ x_dog_n      Y_dog_n ] ] ] ]

```

Рисунок 5 – Листинг программы анимации задачи погони

Для создания анимации в расчёт вставляется график (рис. 6 и 7), в область данных которого вставлена функция *frame* с аргументом *t*, который пользователь связывает с вектором *tFrame* (рис. 6), хранящем номера кадров анимации. Числа 10 и 1000 заменить на другие, регулируя тем самым скорость и длительность анимации⁴.

⁴ Можно показать очередные 24 кадра бега собаки за зайцем, а в следующий кадр вставить какую-нибудь информацию. Например, объявление о продаже борзых щенков, реализуя тем самым миф о двадцать пятом рекламном кадре.

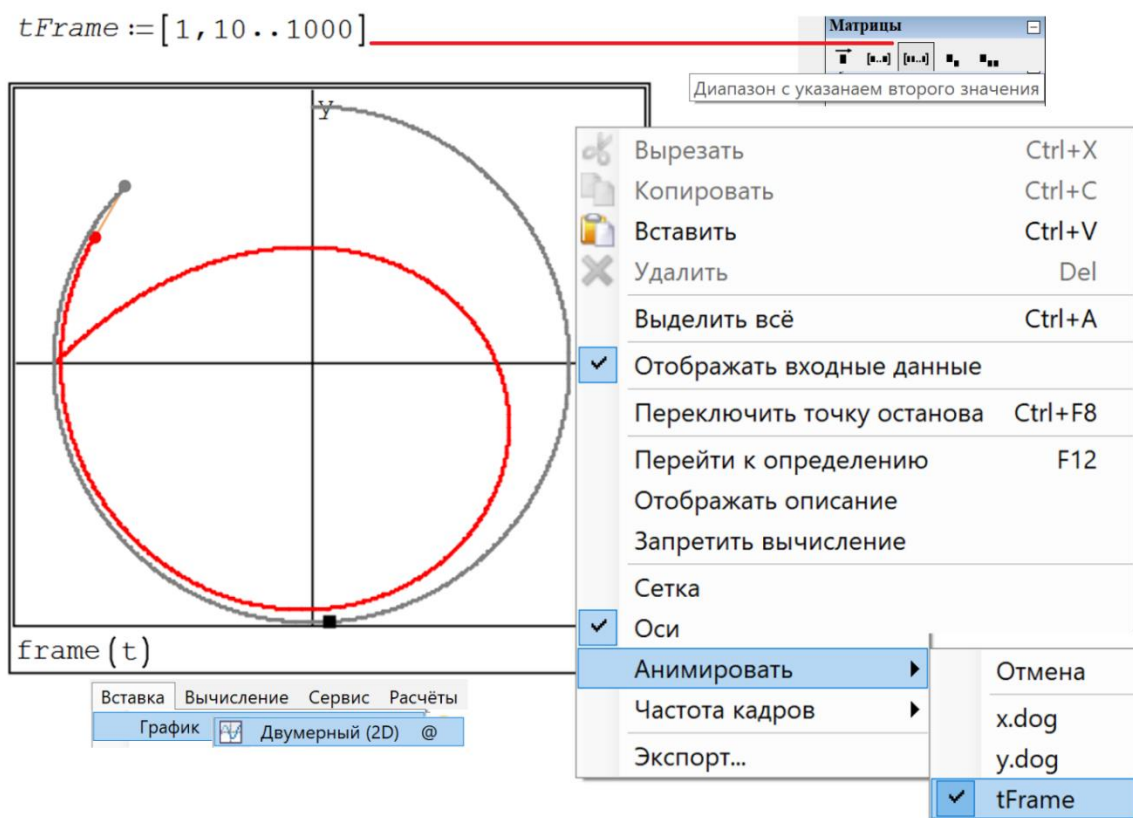


Рисунок 6 – Подготовка к созданию анимации

Частота кадров анимации и другие её параметры задаются также через диалоговое окно, показанное на рисунке 7.

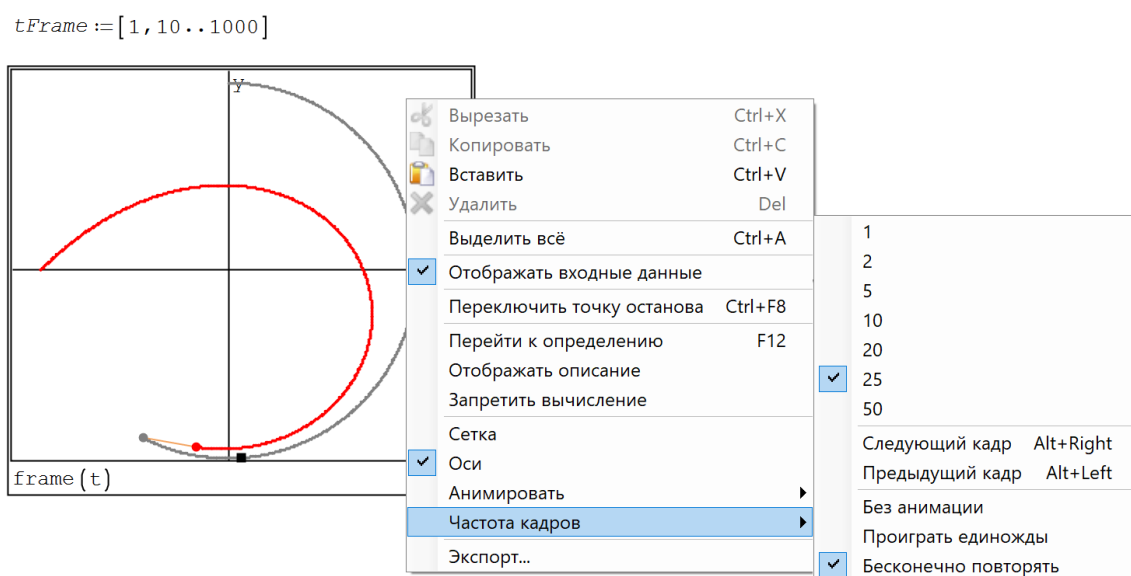


Рисунок 7 – Установка параметров анимации

Команда Экспорт (см. внизу рисунка 7) позволяет созданную анимацию экспортировать в gif-файл и использовать её вне пакета SMATH Studio.

На сайте компании РТС⁵ можно увидеть авторскую анимацию разных случаев погони собаки за зайцем. Там также размещена анимация частного случая задачи о погоне – задача о летучих мышах, которые попарно гоняются друг за другом⁶. Рассмотрены и трёхмерные случаи (рис. 8)⁷.

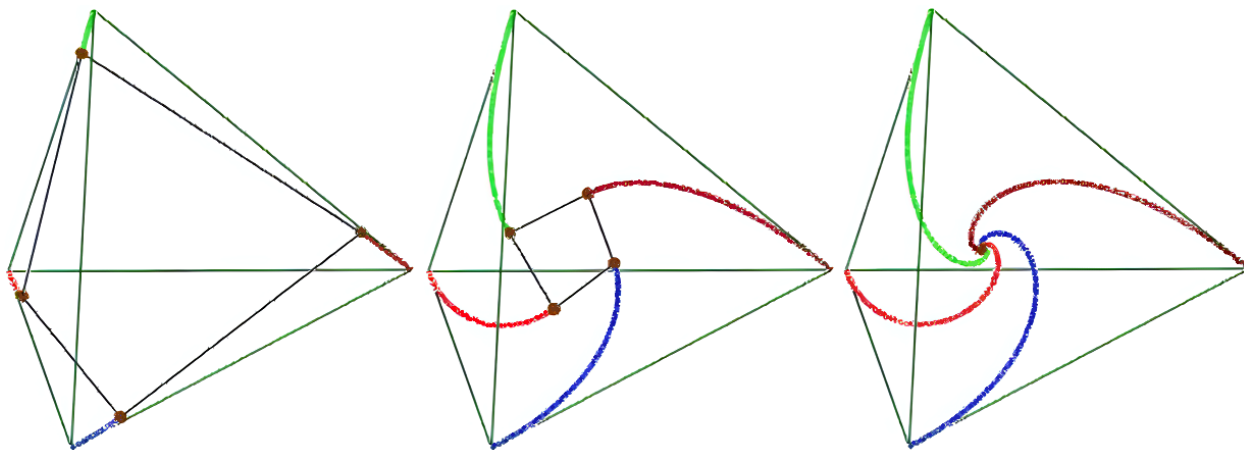


Рисунок 8 – Визуализации задачи, когда летучие мыши вылетают из вершин тетраэдра и гонятся друг за другом: синяя за коричневой, коричневая за зелёной, зелёная за красной и красная за синей (решение Вернера Эксингера)

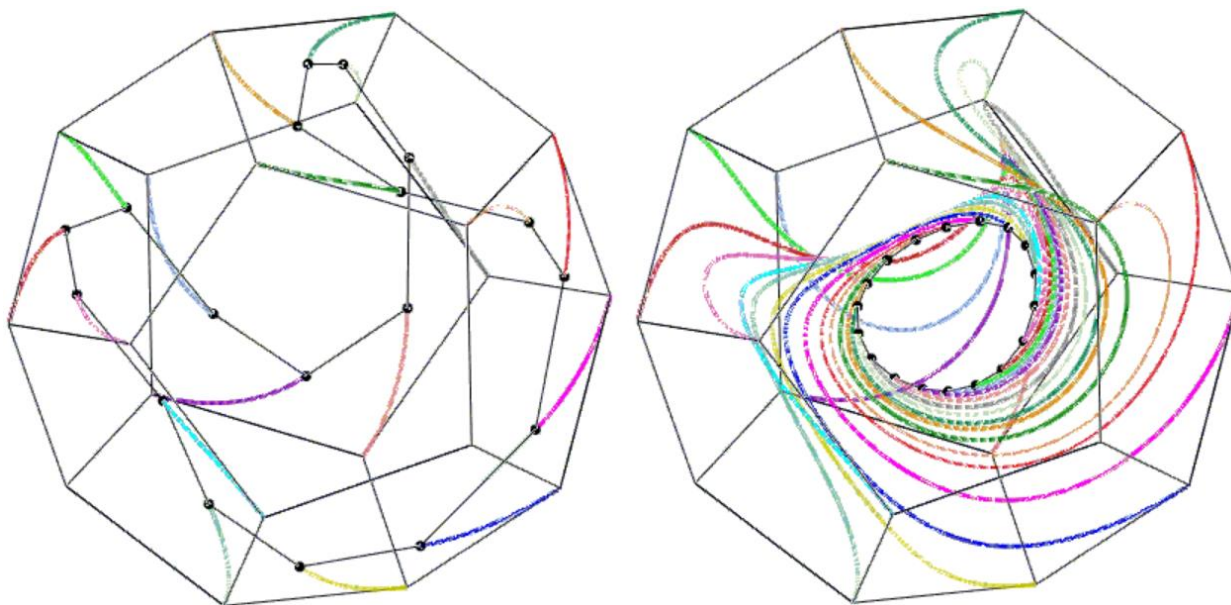


Рисунок 8а – Визуализации задачи, когда летучие мыши вылетают из вершин многогранника (решение Вернера Эксингера)

⁵ <https://community.ptc.com/t5/Mathcad/Wolf-and-hare-one-old-problem-with-simple-Mathcad-solution/m-p/574235>

⁶ <https://community.ptc.com/t5/Mathcad/Mice-problem/m-p/575570>

⁷ <https://community.ptc.com/t5/Mathcad/Is-it-my-own-or-Mathcad-15-error/td-p/576164/page/2>

Способ 2. Была когда-то радиопередача «Разучим песню». В ней часто звучали такие слова: «А давайте теперь прослушаем мелодию этой песни в исполнении другого музыкального инструмента, например, контрабаса!». Затем исполнялся очередной куплет разучиваемой песни в новой аранжировке.

А давайте решим задачу о погоне другим компьютерным инструментом – в среде визуального программирования SimInTech! Сделаем ей новую аранжировку и перекинем мостик от первого названия статьи ко второму.

Честно говоря, новое решение будет походить на стрельбу из пушки по воробьям, пардон, по зайцам. Но мы это сделаем не для решения задачи, а для показа хода решения: «Цель ничто – движение всё!». Кстати, поговорка про пушку и воробья имеет ввиду то, что стреляют ядрами. Но можно из пушки пальнуть по воробьям картечью или дробью, защищая, к примеру, урожай ягод от пернатых налетчиков. В эпиграфе статьи упоминался роман Толстого – «Война и мир» с красочными сценами псовой охоты. Во втором романе Льва Николаевича «Анна Каренина» мастерски описана ружейная охота. Ствольная артиллерия и ракетное вооружение тоже имеют прямое отношение к нашей задаче о погоне в том случае, если снаряды и ракеты управляемые. Оптимальным решением нашей задачи, минимизирующим время полета, будет такое: стреляющий оценивает место встречи летящего и догоняющего объектов, и стреляет по прямой на опережение.

Сразу отметим, что одна из особенностей SimInTech заключается в том, что в решении нет вектора времени и вектора местоположений собаки и зайца, как в решении на SMath Studio, но есть системная переменная *time*, хранящая время, чем мы и воспользуемся.

Расчёт в пакете SimInTech идёт динамически. Когда он стартует, время запускается и движется только вперед как в реальной жизни. Как пела незабвенная Алла Пугачёва: «Жизнь невозможно повернуть назад, / И время ни на миг не остановишь!». Хотя нажать на кнопку Пауза можно.

Поговорим об этих кнопках и о том, как задачу о погоне можно решить в среде SimInTech!

На рисунке 9 показаны три последовательных скана интерфейса SimInTech при решении задачи о погоне: запуск программы (1), открытие нового расчётного проекта (2) и сам новый расчётный проект – модель общего вида (3). Можно выбрать заготовки и других проектов, если планируется решить специфическую задачу из области, например, тепло- или электроэнергетики.

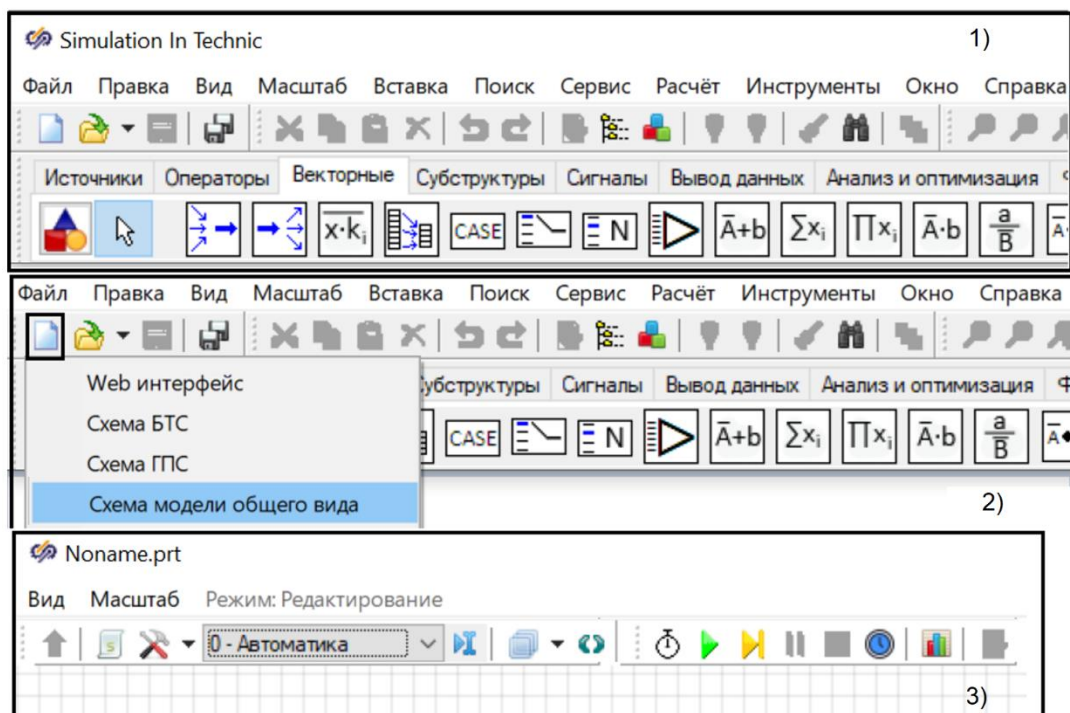


Рисунок 9 – Начало работы в среде SimInTech

Задачу о погоне в среде SimInTech можно решить разными путями, используя разные инструменты из богатого набора. Мы максимально повторим расчёт, сделанный в среде SMath Studio (см. рис. 2 и 3), последовательным выполнением расчётных операторов. Для этого в пустой элемент вставляется блок «Язык программирования» (PL – см. рис. 10) из вкладки Динамические. Для этого достаточно щёлкнуть левой кнопкой мышки сначала по нужной кнопке панели инструментов, а потом в нужное место поля проекта, разлинованного в клеточку. Появится прямоугольник с одним входом слева ($>$), связанным с переменной x , и с одним выходом справа ($>$), связанным с переменной y .

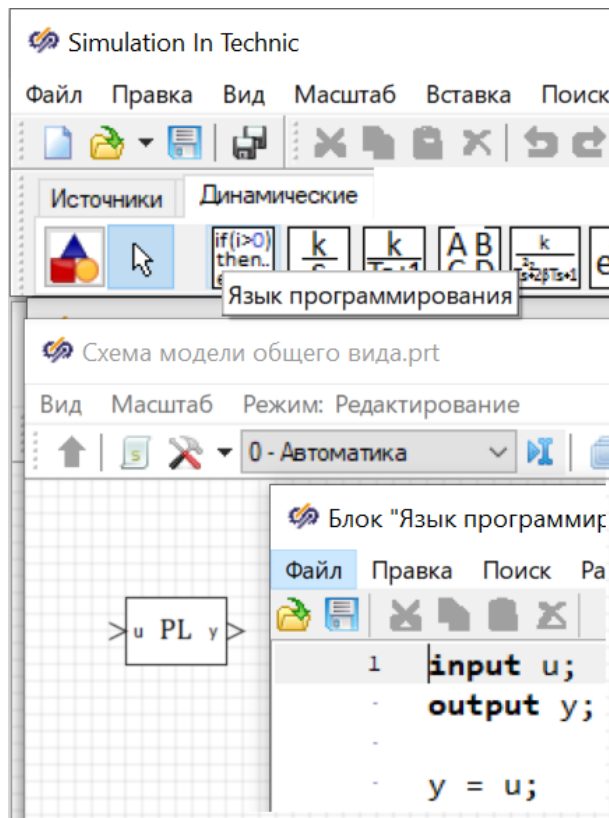


Рисунок 10 – Вставка в расчётный проект программного блока

Если затем два раза щёлкнуть мышкой по блоку PL, то откроется окно с именем «Язык программирования», также показанный на рисунке 10. В нем хранится программа-пустышка, которую можно назвать так – «За что купил, за то и продаю!»: значение входной переменной u без изменений передается выходной переменной y , и всё! Содержимое этого окна нужно будет стереть и записать в нём программу, показанную на рисунке 11 и во многом повторяющую запись на рисунке 2.

```

Блок "Язык программирования": LangBlock_1
Файл  Правка  Поиск  Расчёт  Справка  Инструменты
[Icons]
1 [ ] initialization
-   R = 700;
-   V_dog = 5.35;
-   V_hare = 5;
-   X_dog = 0;
-   Y_dog = 0;
-   fi = 0;
-   end;
-   D_alfa = V_hare*stepsize/5/R;
10 X_hare = R*sin(D_alfa*time);
-   Y_hare = R*cos(D_alfa*time);
-   fi = atan2(Y_hare - Y_dog, X_hare - X_dog);
-   X_dog = X_dog + V_dog*stepsize/5*cos(fi);
-   Y_dog = Y_dog + V_dog*stepsize/5*sin(fi);
-   s = sqrt((X_dog - X_hare)^2 + (Y_dog - Y_hare)^2)
-   |
-   output X_dog, X_hare, Y_dog, Y_hare, s;
initialization
en|
1 [ ] initialization
2 | end;|

```

Рисунок 11 – Листинг программа решения задачи о погоне в среде SimInTech

Область программы с заголовком *initialization* требует небольших пояснений, которые даны справа: вручную набирается соответствующее ключевое слово, далее делается переход на новую строку и пишется слово *end*, завершающееся точкой с запятой. После этого появятся две вертикальные линии, одна из которых имеет вверху квадратик с минусом. Если щёлкнуть мышкой по этому квадратику, то область под ним свернется, а минус превратится в плюс. Повторный щелчок мышкой по квадратику разворачивает область, в которую нужно будет записать константы (радиус круга, по которому бежит заяц, скорость его и собаки), а также стартовые координаты собаки и её начальный азимут ($f_i - \varphi$), для которого сгодится и нуль. Этот нуль, кстати, можно написать и в расчёте на рис. 2 вместо довольно громоздкой формулы с арктангенсом. На ход расчёта это упрощение почти не влияет – первый шаг бега собаки будет несколько иным.

Кстати, в среде SMath Studio часть расчёта также можно спрятать в область, которую допустимо дополнительно запереть паролем. Эта возможность была бы нелишней и для пакета SimInTech: заказчику отдается расчёт, где некоторые фрагменты (ещё не запатентованные know-how, например) раскрываются только за дополнительную плату.

Имена переменных и функций (идентификаторов) в программе на рисунке 11 раскрашены⁸ автоматически: чёрный цвет – пользовательские конструкции, синий – числовые константы, красный – системные переменные и коричневый – встроенные функции.

В нашей программе две системные переменные, значения которых задаются не пользователем, а самой системой – пакетом SimInTech. Это переменные **stepsize** и **time**. Первая – это шаг интегрирования, а вторая – модельное время, о чем поведала справка <https://help.simintech.ru>. Не будем вдаваться в суть этих переменных – программа работает и хорошо. Пришлось только переменную **stepsize** делить на 5, чтобы анимация бега зайца с волком была не такой быстрой.

После ввода показанной на рисунке 11 программы в блок PL он преобразуется в вид, показанный на рис. 12: входной порт пропал (нет слов input), но появились пять выходных портов (>). Программа без входных данных будет самодостаточной, если так можно выразится. Исходные данные (скорости зайца и собаки, а также стартовые координаты собаки) вводятся через редактирование программы. То можно добавить в блок PL входные порты (один из них показан на рисунке 10) и водить исходные данные через них. Но мы не будем усложнять проект!

На рисунке 12 показано, как в наш расчётный проект к блоку PL добавлены ещё четыре блока: два блока-мультиплексора (собирателя входных сигналов в один выходной сигнал), и два блока построения графиков. Первый график – это график (анимация) погони собаки за зайцем (рис. 13 – фазовый портрет), а второй – это график изменения во времени расстояния собаки от зайца (рис. 14 – временной график). После ввода описанных четырех блоков через нажатие соответствующих кнопок на панели инструментов Векторные и Вывод данных и соединения мышкой портов вывода с портами ввода так, как это показано на рисунке 12, проект будет почти готов. Нужно будет нажать на кнопку Пуск (зелёный треугольник с прямым углом, направленным вправо), а далее двойным щелчком мыши по блокам-графиков открыть сами графики и отформатировать их так, как показано на рисунках 13 и 14. Мы не будем описывать специфику такого форматирования, а скажем лишь что щелчком правой кнопки мыши по нужному блоку можно открыть соответствующие диалоговые окна, где «кто ищет, тот всегда найдет». Если же поиск будет

⁸ На заре эры персональных компьютеров первый автор послал в журнал «Микропроцессорные средства и системы» (он выпускался с 1984 по 1990 г.) статью с идеей раскраски листингов программ. Статью отклонили под предлогом того, что цветные дисплеи и принтеры у нас в СССР ещё не скоро появятся. Кроме того, рецензент честно признался в том, что он дальтоник, и цвет в программах вызывает у него только рябь в глазах. А примерно в то же время появился язык программирования Turbo Pascal с цветом в программах. Подробнее об этом – см. пункт 129 здесь <http://tw.t.mpei.ac.ru/ochkov/EEE-2-2022-My-Energy.pdf>.

неудачным, то поможет справка по пакету или поиск в интернете по грамотно составленному ключевому слову, что приведет в том числе и к обучающим видеороликам на YouTube и RuTube.

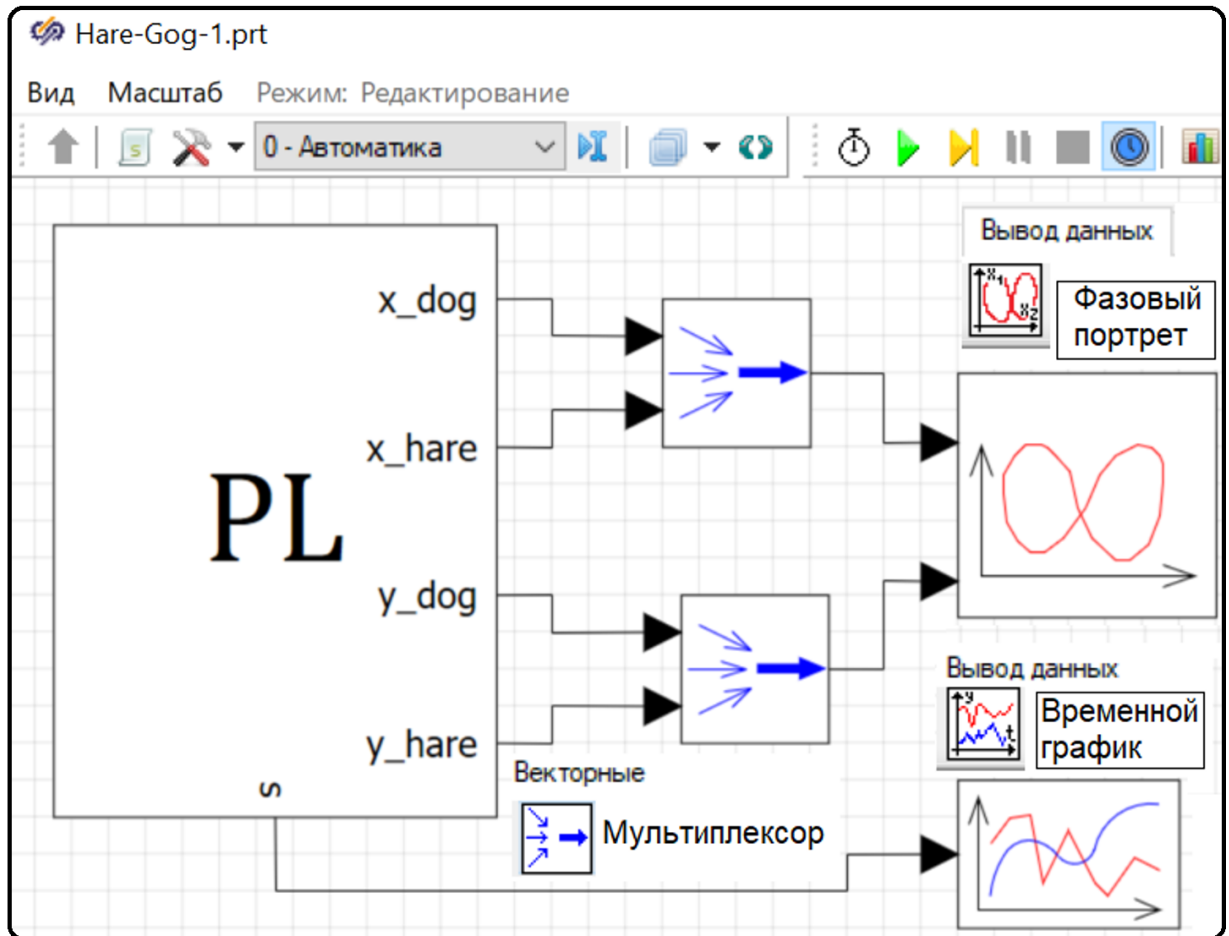


Рисунок 12 – Готовый проект «Погоня собаки за зайцем»

Рисунки 13 и 14 – это не просто графики, фиксирующие момент поимки зайца собакой, а финальные кадры двух анимаций, которые будут появляться около окна проекта (рис. 12) после того, как будет нажата кнопка Пуск. Четыре кадра такой анимации погони показаны на рисунке 3.

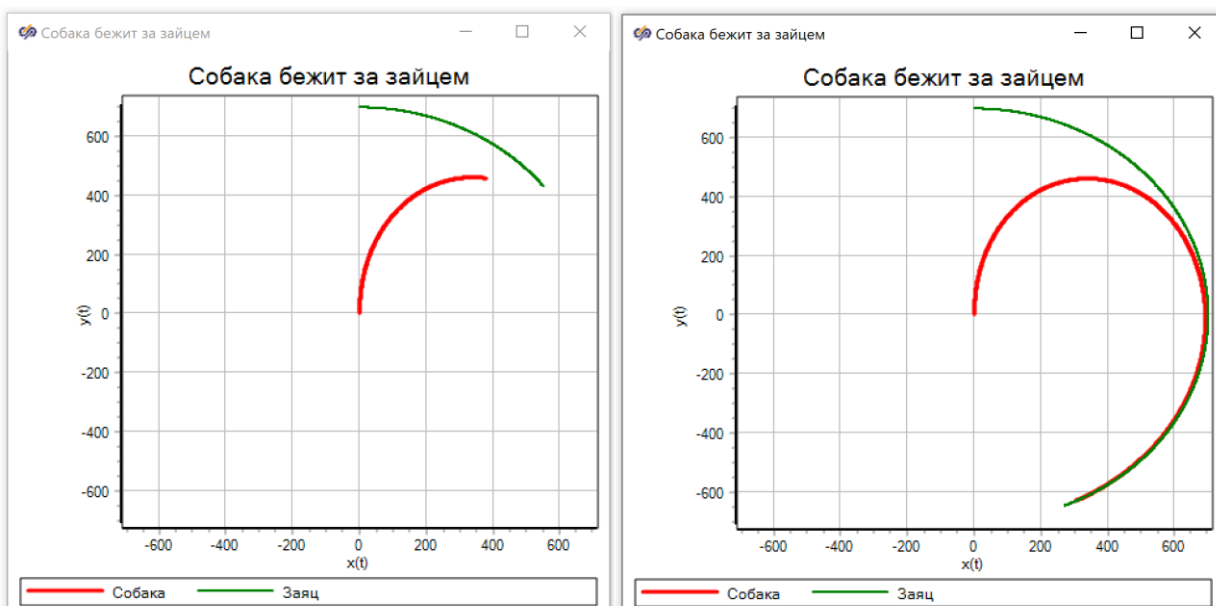


Рисунок 13 - Два кадра анимации погони собаки за зайцем в SimInTech

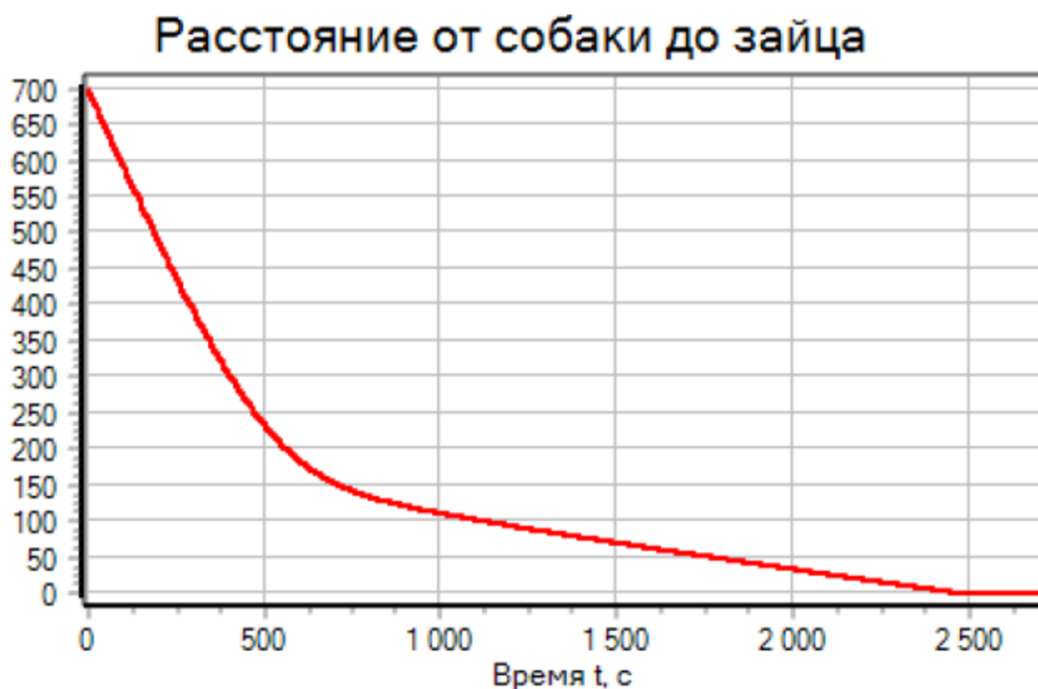


Рисунок 14 – График изменения расстояния от собаки до зайца в процессе погони

Если сравнивать рисунки 2 и 11, то можно отметить следующие моменты:

1. Решение в среде SMath Studio оперирует физическими величинами с единицами измерения, что является современным трендом, а где-то и стандартом.
2. Записи в среде SMath Studio повторяют записи, сделанные в обычных книгах, справочниках и статьях – корень, есть корень, степень, есть степень, дробь

есть дробь. Решение же в среде SimInTech – это некая китайская грамота, доступная для понимания только избранным – тем, кто знаком с языками программирования, зарождавшимися ещё в прошлом веке.

3. В именах переменных и функций пакета SMath Studio допустимы нижние индексы (мы о них писали), русские и греческие буквы.

В настоящее время ведется работа по внедрению стиля решения задач пакета SMath Studio в пакет SimInTech, что повысит его производительность и удобство программирования, а также избавит от ряда ошибок. В недалёком будущем можно будет увидеть в окне Язык программирования пакета SimInTech (см. рис. 11), не только операторы численной, но и символьной математики SMath Studio, инструменты решения систем уравнений – алгебраических и дифференциальных и много другого из арсенала SMath Studio.

Кратко о второй части названия статьи.

В Московском энергетическом институте (НИУ «МЭИ») пакеты SMath Studio⁹ и SimInTech в их лицензионных версиях широко используются не только для обучения студентов, но и для создания цифровых двойников энергообъектов для создания тренажёров, например. А в демо-хранилище SimInTech можно найти много чего интересного в этом отношении. Так на рисунке 15 показана «вершина айсберга» цифрового двойника турбоустановки с такими главными составными частями: компрессор, камера сгорания и газовая турбина. В энергетике к подобным турбоустановкам на вал цепляют электрогенератор, а выхлопные газы направляют в котёл, пар из которого идет в паровую турбину с ещё одним электрогенератором. За счёт такой бинарной парогазовой схемы резко повышается КПД энергоустановки. Значение КПД тепловой машины, как известно, зависит от температур горячего и холодного источника тепла. Газовая турбина повышает температуру горячего источника, а паровая турбина – холодного.

⁹ Установки для трансформации тепла и охлаждения: расчеты на SMath : учебное пособие для вузов / Н. Л. Бударин, А. В. Мартынов, В. Ф. Очков [и др.]. — 2-е изд., стер. — Санкт-Петербург : Лань, 2025. — 184 с. — ISBN 978-5-507-52118-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/438536> (дата обращения: 21.01.2025). — Режим доступа: для авториз. пользователей.

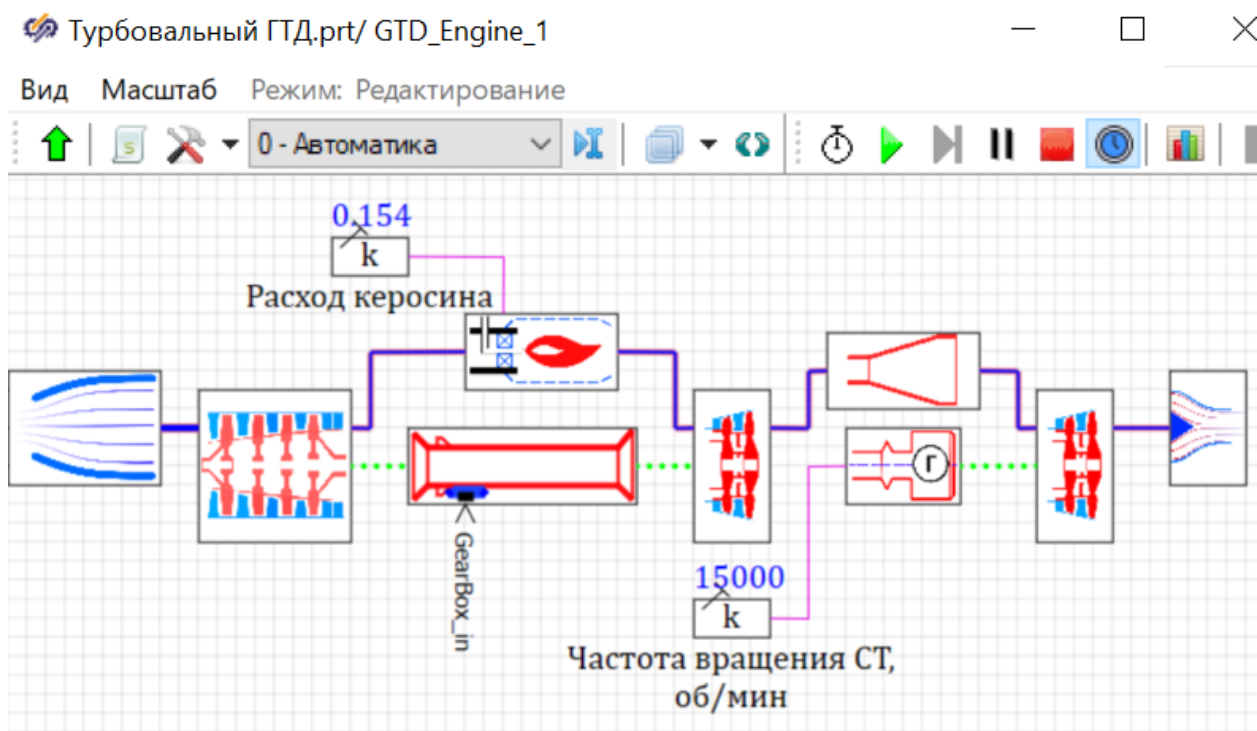


Рисунок 15 – Цифровой двойник авиационного турбореактивного двигателя

Подводную часть «айсберга» можно увидеть, если щёлкнуть мышкой по отдельным составляющим проекта, показанного на рисунке 15. Будут открываться цифровые двойники отдельных частей и элементов газотурбинной установки.

Авиационный турбореактивный двигатель здесь оказался не случайно. Некоторые БПЛА самолётного типа используют такие двигатели. А сбивают их управляемыми ракетами, летящими по траекториям, описанным в этой статье.

На сайте¹⁰ в пункте 13 хранятся файлы расчётов, показанных выше, а также альтернативные решения.

Задания читателю:

1. Скачайте в интернете программные средства, описанные в статье, и решите поставленные задачи.

2. На рисунке 16 показаны два кадра анимации (один из начальных и один из конечных) погони собаки за зайцем, бегущим не по кругу (рис. 3 и 4), а по квадрату. Реально такой случай реализовать довольно сложно, а виртуально – нет проблем. Более того, можно заставить бежать зайца по периметру треугольника и любого многоугольника, и в общем случае по любой кривой.

¹⁰ <http://www.twt.mpei.ac.ru/ochkov/16MIT>

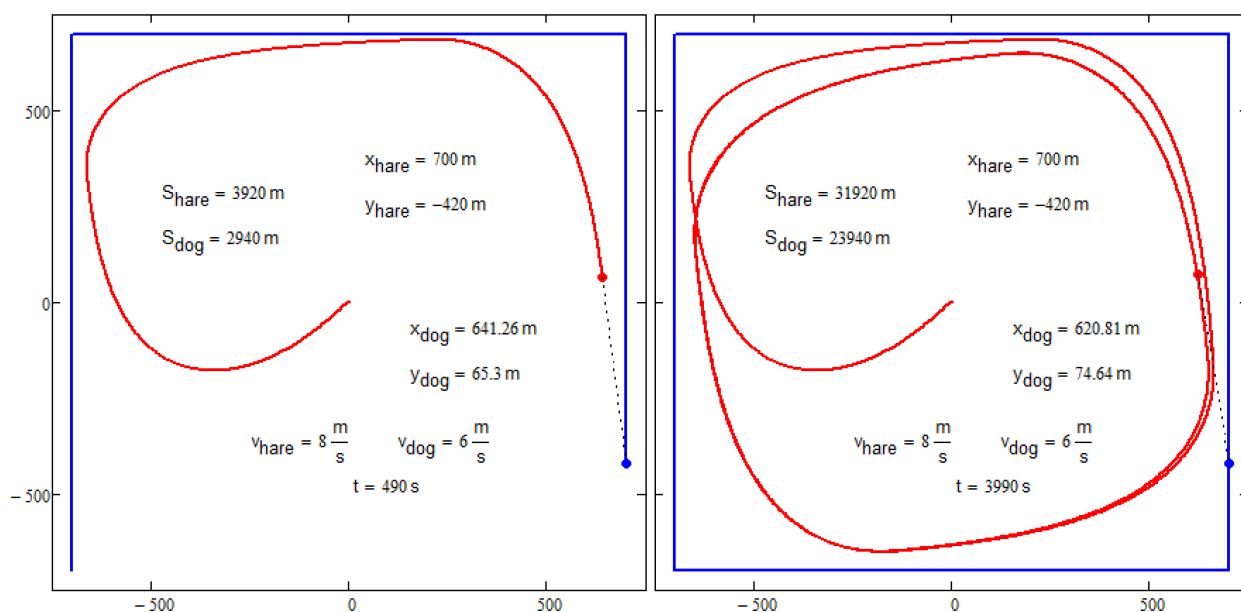


Рисунок 16 – Погоня собаки за зайцем, бегущим по квадрату

Так вот, найдите, дорогой читатель, аналитическую запись формы замкнутой кривой, к которой асимптотически стремиться собака. Скорость собаки меньше или равна скорости зайца.

3. Если собака будет стартовать не из центра, а из какой-то другой точки, то погоня закончится либо поимкой зайца, либо выбегом собаки на замкнутые бесконечные траектории, показанные на рис. 4 и 16. Читатель, раскрасьте область точек старта собаки в один цвет, если собака догонит зайца. Ежели этого не случится, то цвет места старта собаки будет другим. Подсказку к решению можно найти в статье¹¹. Должно получиться интересное произведение изобразительного искусства, которое можно повесить на стене комнаты, показывать его удивлённым гостям, снимать со стены, переворачивать на обратную сторону и демонстрировать математическую суть рисунка – некоторые кривые из данной статьи.

¹¹ Очков В.Ф., Чудова Ю.В., Умирова Н.Р. Портрет корней системы уравнений // Математическое образование № 3 (103), 2022. С. 33-46 (<http://twf.mpei.ac.ru/ochkov/Portrait-Roots.pdf>).