

Тело, брошенное под углом к горизонту

ЗАДАНИЕ 2.1

- 1) Записать векторное уравнение движения тела и уравнения движения в проекциях на декартовы оси x и y .
- 2) Открыть окно новой модели **Simulink** и создать подсистему (Subsystem) с 4-мя входными портами (In) и тремя выходными (Out).
- 3) Задать с помощью блоков констант (Constant) значения входным параметрам.
- 4) К выходным портам подсистемы подсоединить блоки для визуализации результатов из библиотеки Sinks.
- 5) Войти в подсистему двойным нажатием левой кнопки мыши и собрать схемы интегрирования дифференциальных уравнений относительно x и y .
- 6) Выполнить моделирование движения тела, брошенного под углом к горизонту, проанализировать графики, определить по ним максимальную высоту и дальность полета, а также время полета и время достижения максимальной высоты.
- 7) Изменить начальный угол броска и повторить запуск модели. Сделать выводы о влиянии угла на дальность приземления тела.
- 8) Изменить массу, коэффициент сопротивления и начальную скорость, сделать заключение, какой из параметров сильнее влияет на дальность полета.
- 9) Составить «Отчет о работе», куда включить:
 - уравнения движения,
 - блок-схемы для решения системы дифференциальных уравнений.
 - графики.
 - выводы.

ВСПОМОГАТЕЛЬНЫЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ ЗАДАНИЯ, А ТАКЖЕ СХЕМЫ И ГРАФИКИ, ПОМЕЩАЕМЫЕ В ОТЧЕТ ПО РАБОТЕ

- 1) Векторное уравнение движения тела и его проекции на декартовы оси x и y

$$m\vec{a} = \vec{G} + \vec{R}_{\text{сопр}};$$

$$m\ddot{x} = -k\dot{x};$$

$$m\ddot{y} = -mg - k\dot{y}.$$

- 2) Создание подсистемы (Subsystem) в окне новой модели Simulink.

Чтобы создать входы и выходы в блоке Subsystem, скопированном из библиотеки Ports & Subsystems в окно новой модели Simulink, требуется дважды кликнуть по блоку Subsystem, разорвать связь между портами (In) → (Out), скопировать или перетащить из библиотеки Ports & Subsystems нужное количество входных и выходных портов и сделать под ними соответствующие подписи. Те же подписи появятся также на головном блоке подсистемы.

4 входных (In) порта:

(m) - масса тела,
(k) - коэффициент сопротивления воздуха,
(V_0) - начальная скорость брошенного тела,
(u_0) - угол броска по отношению к горизонту,

3 выходных (Out) порта:

координата x ,
координата y ,
проекция скорости на вертикальную ось V_y .

3) Задание значений входных переменных.

Перетаскиваем мышью из библиотеки Commonly Used Blocks блоки констант (Constant), подсоединяем их ко входам подсистемы и задаем в них значения для входных параметров. Указываем под каждым из блоков соответствующее название с размерностью:

$m_{kg}=2$;
 $k_{Ns/m}=0.06$;
 $V_{0\ m/s}=850$;
 $u_{0\ deg}=45$.

4) Получение визуальной информации о выходных переменных

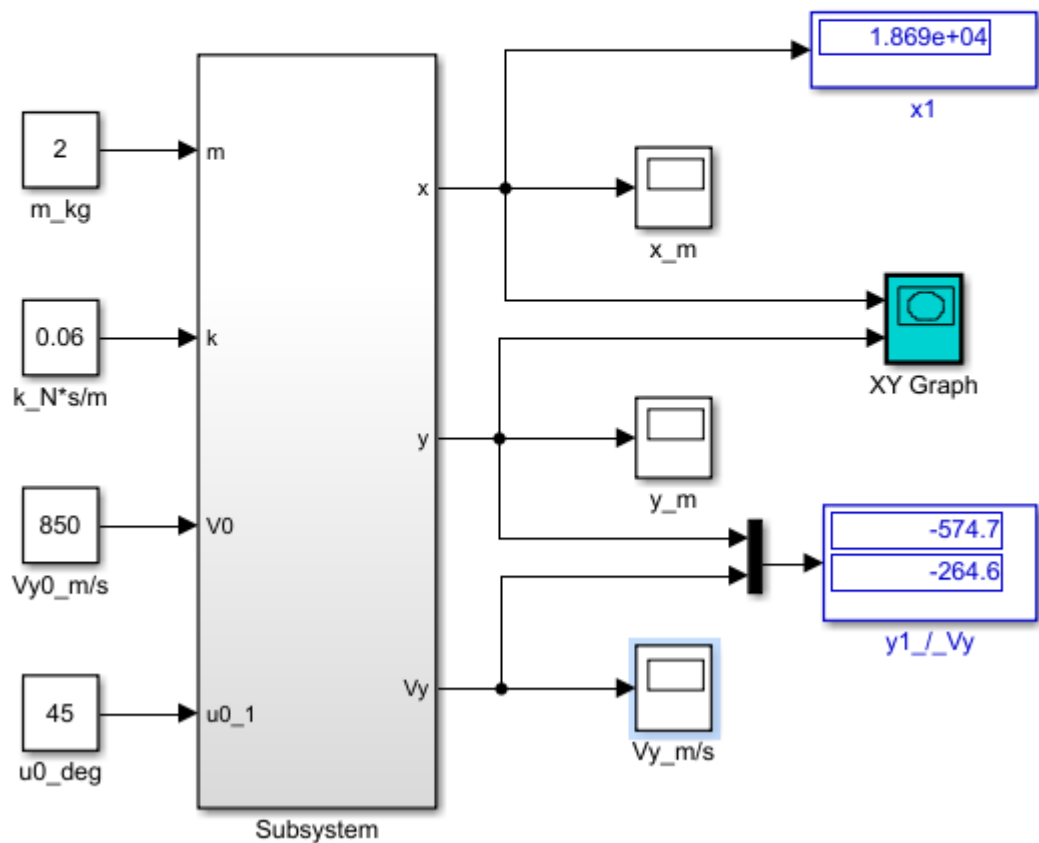
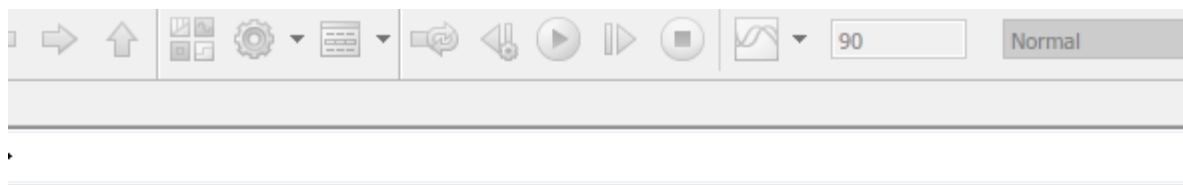
К выходным портам подсистемы подсоединяем блоки из библиотеки Sinks :

цифровой дисплей (Display),
осциллограф (Scope),
графопостроитель (XY Graph).

Делаем подписи под всеми блоками. Для одновременного вывода на один цифровой дисплей значений координаты y и скорости V_y добавляем мультимплексор (Mux), который обеспечивает передачу на один выход нескольких входных сигналов.

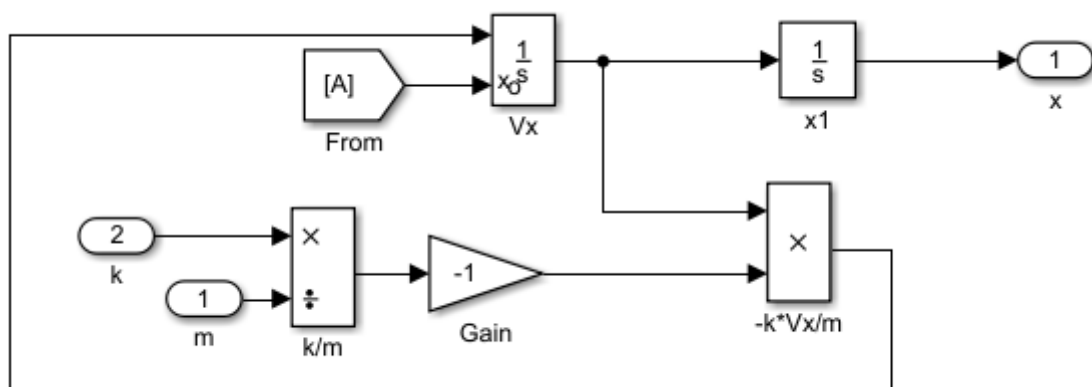
Цвет блоков можно менять, кликая по ним правой кнопкой мыши и выбирая пункты из раздела Format.

Время моделирования по умолчанию равно 10 сек (оно отображается в окошке на верхней панели). За такое время вся траектория полета построена не будет. Для вывода всей траектории при заданной скорости броска (850 м/с) изменим время моделирования на 90.

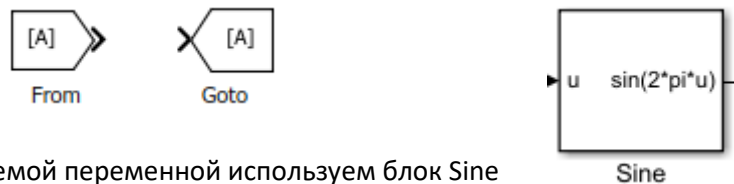


5) Внутреннее содержание подсистемы

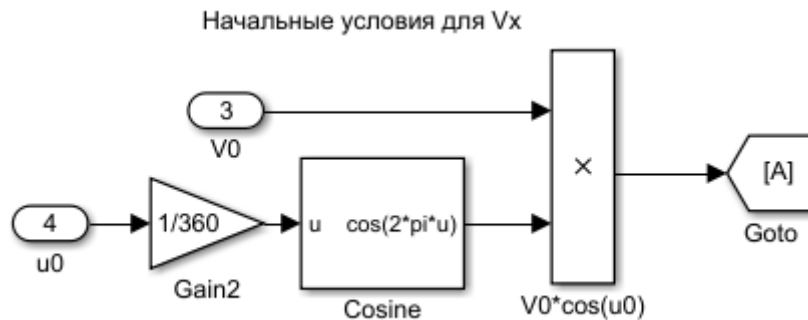
Войдем в подсистему двойным нажатием левой кнопки мыши и соберем схему интегрирования дифференциального уравнения относительно x .



Составляем отдельную схему вычисления $V_{x0} = V_0 \cdot \cos(u_0)$ – начального значения проекции скорости на x – и передаем ее без участия соединительных линий на вход первого интегратора с помощью блоков **From** и **Goto** из библиотеки **Signal Routine**:



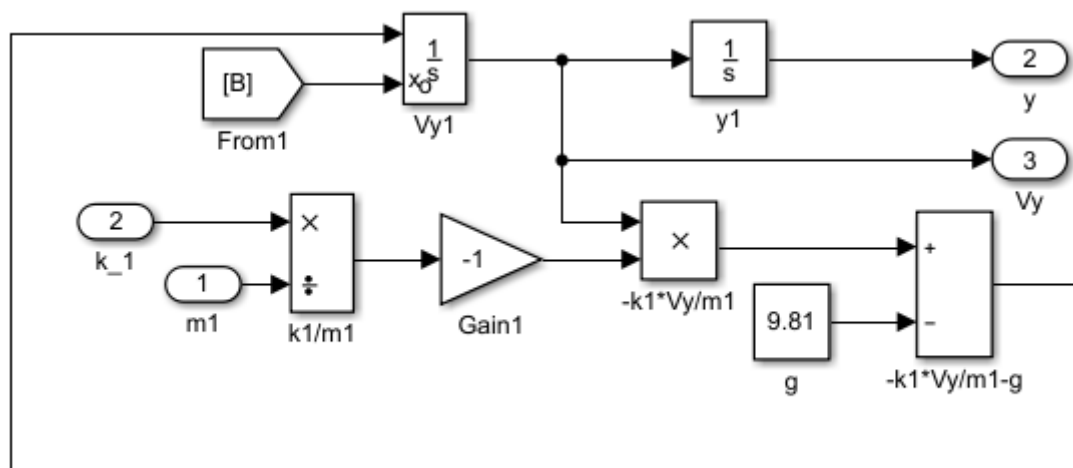
Для вычисления косинуса от задаваемой переменной используем блок Sine библиотеки Lookup Tables, в котором изменяем внутренний параметр в окне «Выводимая формула», выбирая в перечне функций $\cos(\dots)$



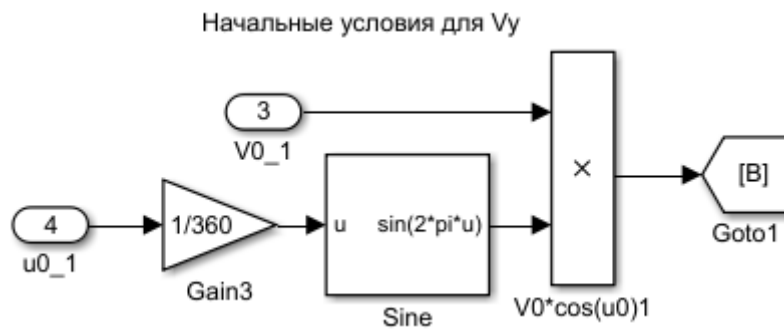
Начальное условие для второго блока интегрирования ($x_0=0$) специально указывать не требуется, т.к. нулевое значение задано по умолчанию.

Под всеми блоками делаем подписи, помогающие отслеживать математические действия. Поскольку названия блоков в Simulink должны быть уникальными, то при одинаковых подписях программа будет указывать на повтор обозначений. В таком случае достаточно либо изменить символ ($A \rightarrow B$), либо добавить числовой индекс ($x \rightarrow x1$).

- 6) Для интегрирования уравнения по y , во многом сходного с дифференциальным уравнением для x , копируем уже составленную схему, вводим блок Add для добавления $-g=-9.81$, изменяем подписи, обозначение параметра (B) блоков From и Goto.




Меняем параметры тригонометрического блока для вычисления $Vy_0 = V_0 \sin(u_0)$ – проекции начальной скорости на ось y



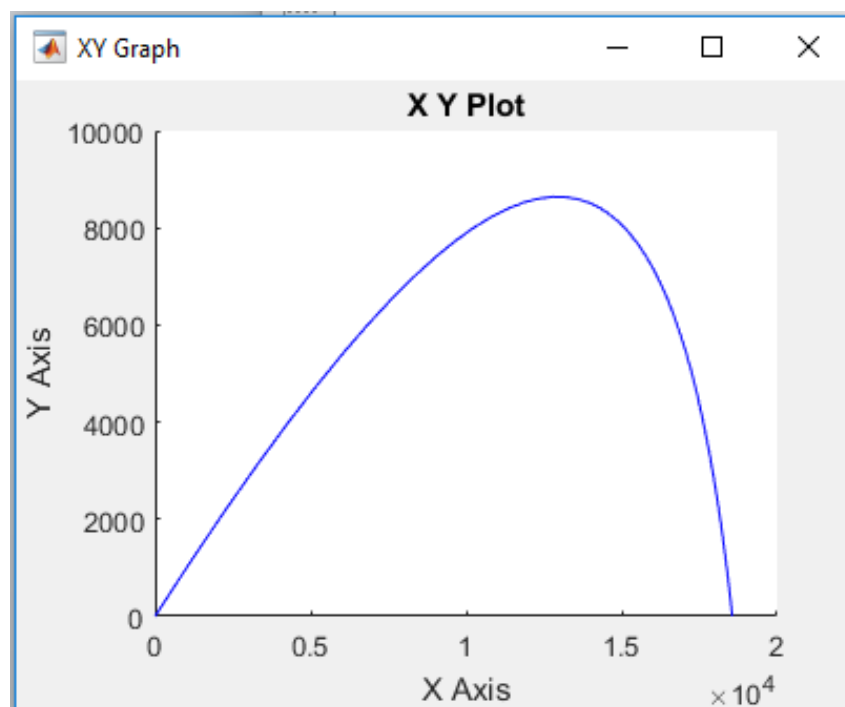
ВНИМАНИЕ! Чтобы в новой схеме использовались прежние значения k и m и в головном блоке Subsystem не появились дополнительные входы, следует удалить скопированные входные порты ($k1$ и $m1$) и еще раз их вставить, **задавая кнопкой мыши действие *Past Duplicate Inport*** (при нажатии клавиш **Ctrl+V** выполняется **Past**). Можно заметить, что в этом случае номера портов сохраняются (порты 2 и 1). Аналогично поступаем с начальной скоростью $V0_1$ и начальным углом броска тела $u0_1$ (входные порты 3 и 4).

7) Процесс моделирования и анализ результатов

Запускаем моделирование, задавая в окне панели инструментов время моделирования 90 (вместо 10 по умолчанию) и нажимая зеленую круглую кнопку **Run** .

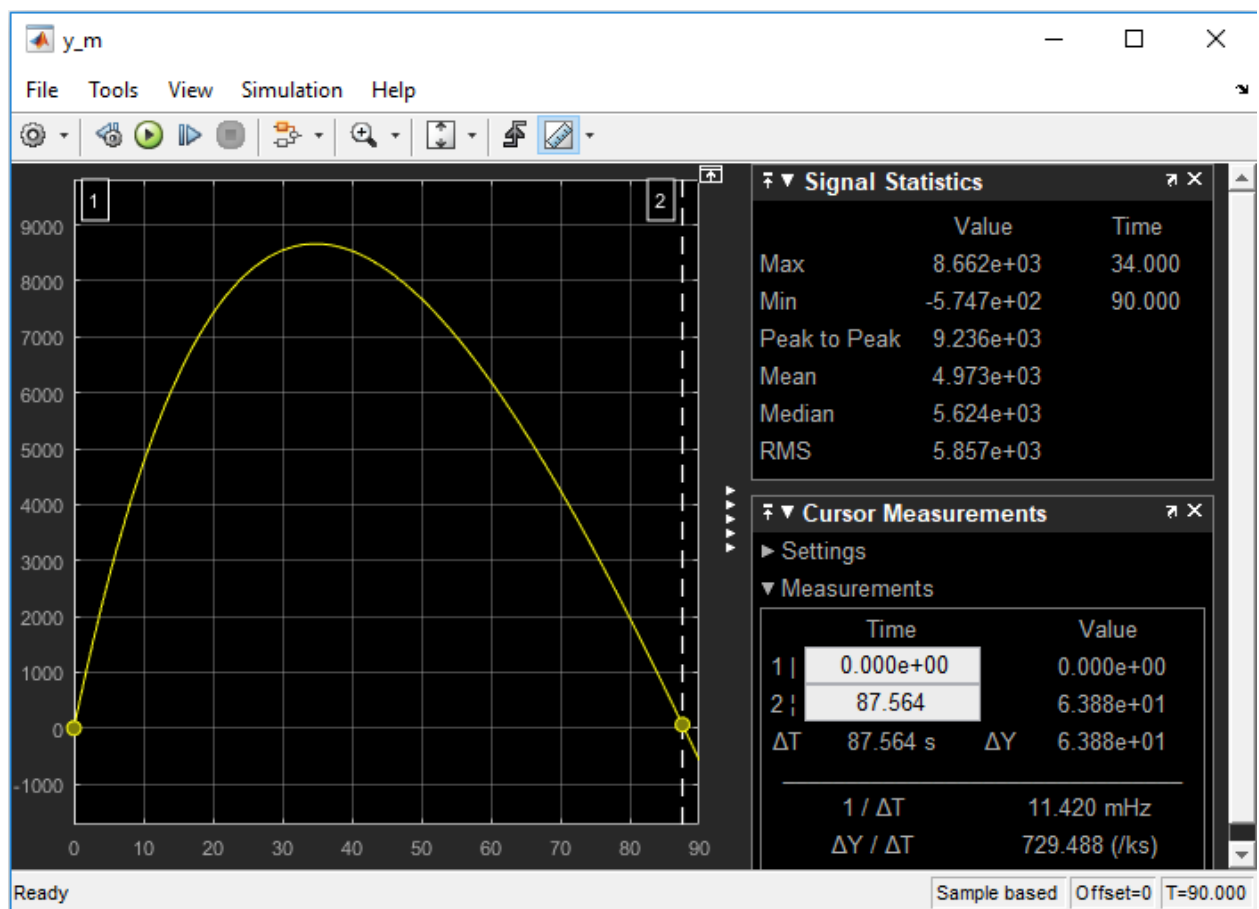
Получаем следующие графики:

Траектория движения



а) График траектории полета тела (графопостроитель XY Graph). Размерность – метры.

График движения по вертикали



б) График зависимости $y(t)$ (осциллограф Scope y_m). Размерность – метры.

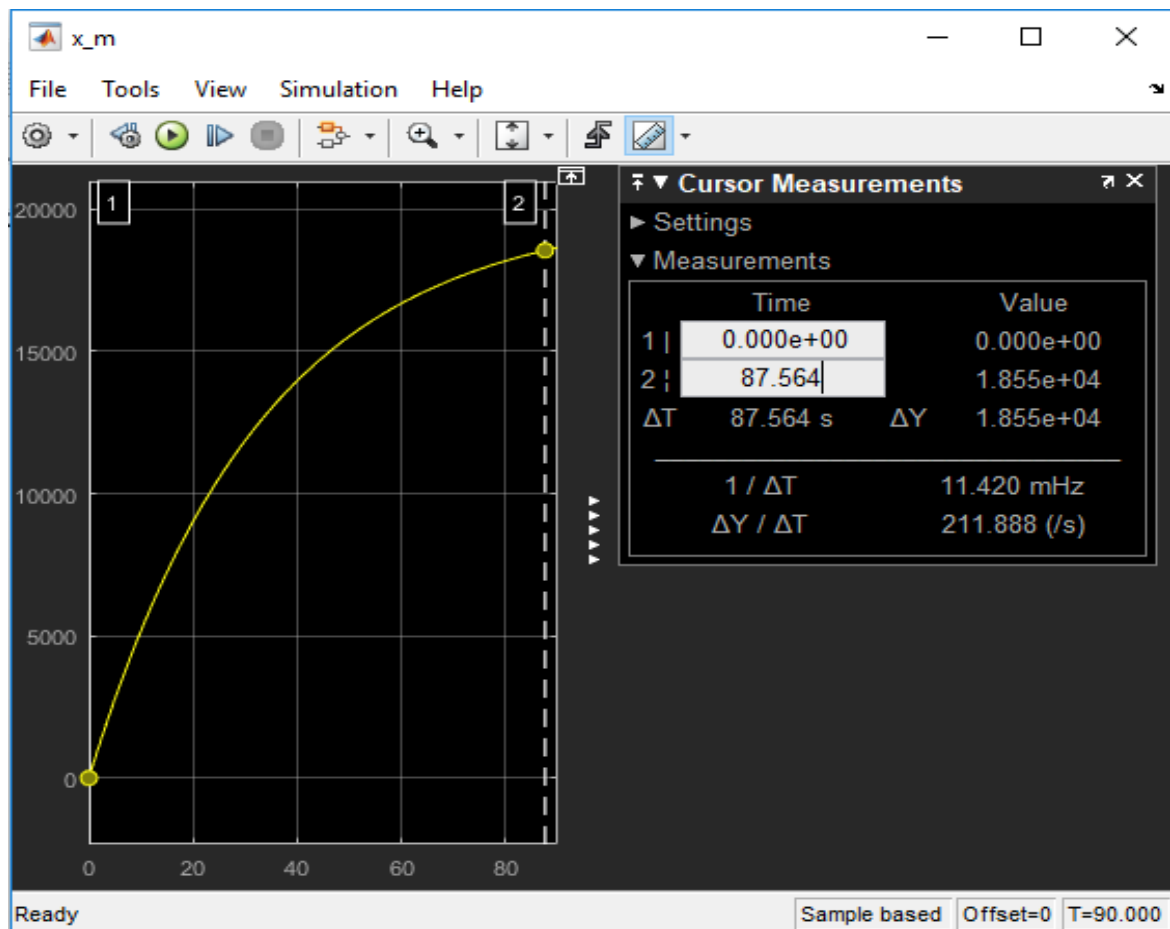
По графику $y(t)$ можем определить время полета, а также максимальную высоту полета и время ее достижения либо используя опции изменения масштаба (zoom+), либо обращаясь к собираемой программой статистике. На верхней панели полученного графика нажимаем **Tools > Measurements > Signal Statistics**. Дополнительно можно нажать крайний правый значок (линейка) для измерения времени движения между заданными положениями.

В результате рядом с графиком откроются две панели данных.

На панели **Signal Statistics** для $y(t)$ в первой строке показана максимальная высота полета (**Hmax=8662 м**) и время достижения этой высоты (**t_Hmax =34 с**);

На панели **Cursor Measurements** для двух точек графика, отмеченных на «0»-м уровне по высоте (0.000e+00), можем определить временной интервал между ними, т.е. время полета $\Delta T=87.564$ s.

График движения по горизонтали



в) График зависимости $x(t)$ (осциллограф Scope x_m). Размерность – метры.

Нажав на крайний правый значок (линейка) на верхней панели построенного графика и открыв окно **Cursor Measurements**, в столбце Time набираем полученное из предыдущего графика время полета ($\Delta T=87.564$ s), рядом в правом столбце Value читаем величину расстояния $1.855e+04$ (в метрах). На данном графике – это расстояние, откладываемое по вертикали, но для нас – это расстояние, на которое тело переместилось по горизонтальной оси координат x . Таким образом, мы определяем, что дальность полета до падения – 18 км 550 м.