Моделирование механических систем в пакете MATLAB Simulink

MATLAB *— высокоуровневый язык технических расчетов, интерактивная среда разработки алгоритмов и современный инструмент анализа данных.*

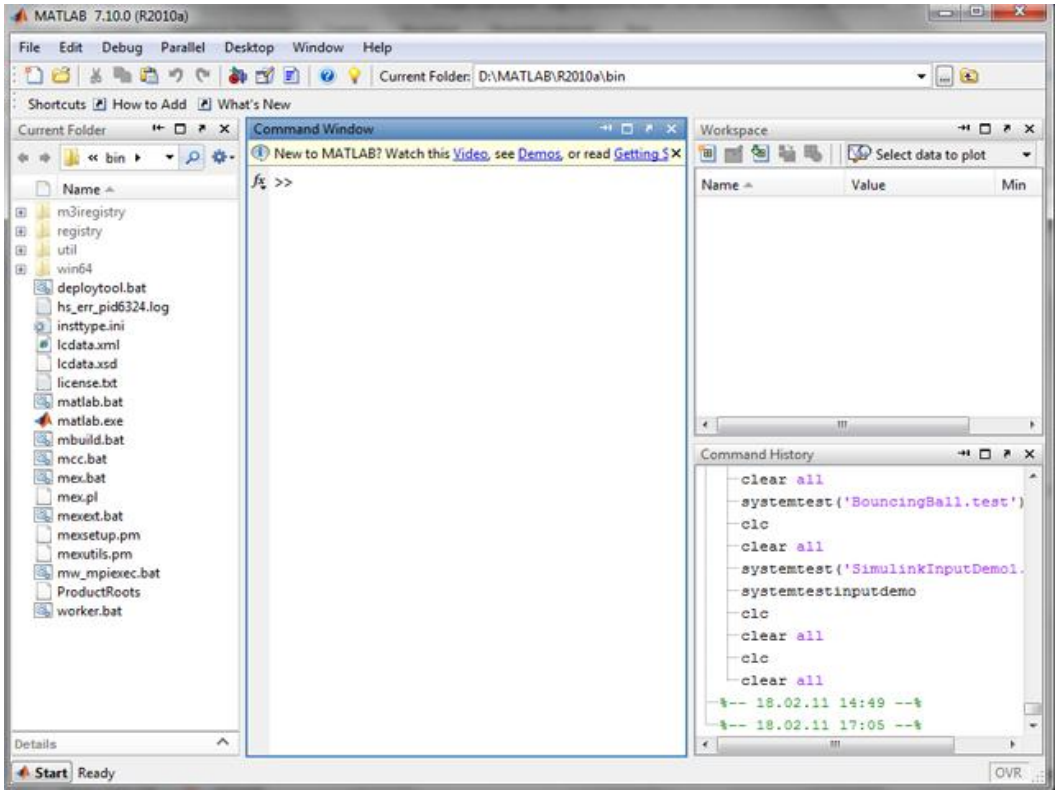
Simulink *- графическая среда имитационного моделирования, позволяющая при помощи блок-диаграмм в виде направленных графов, строить динамические модели, включая дискретные, непрерывные и гибридные, нелинейные и разрывные системы.*

Simscape *— это основная библиотека* ***Simulink*** *для моделирования объектов различной физической природы. Позволяет создавать модели гибридных мультидоменных объектов в виде принципиальных схем, элементов и соединений, реальных физических величин с учетом единиц измерения.*

Simscape *служит основой для моделирования в* ***Simulink*** *электросиловых, механических и гидравлических объектов. Библиотека расширяется специализированными пакетами* ***SimMechanics***, ***SimDriveline***, ***SimHydraulics*** *и позволяет строить модели сложных гибридных мультидоменных объектов для различных задач анализа, в том числе для разработки цифровых систем управления.*

Начало работы в Simulink

При запуске MATLAB перед нами возникает примерно следующее окно:



Пока что не понятно, как здесь можно что-то сделать, однако это очень полезное окно. К нему мы обязательно вернёмся немного позже, а пока сразу приступим к делу. А первым делом нужно выбрать папку, в которой мы будем работать.

Нажимаем на троеточие справа от окошка «Current Folder» и выбираем (создаём) первую понравившуюся папку. Наверняка это будет что-то типа C:\Вася\временное11\123\...

Сразу хочу отметить, что самое интересное в Матлабе и Симулинке – это поиск ошибок. За ним можно проводить ни один месяц, и даже год. Однако ошибки ошибкам рознь... Можно долго и упорно собирать какую-нибудь сложную модель, собрав её, попытаться сохраниться и получить окошко, где в нескольких предложениях описывается содержание ошибки и даже даётся ссылка на более подробный отчёт... В этом случае можно несколько раз пересобрать схему, переустановит матлаб, снести винду, но всё будет бесполезно... А на деле оказывается, что Вы где-то в окне Симулинка просто поставили русскую букву или слово.

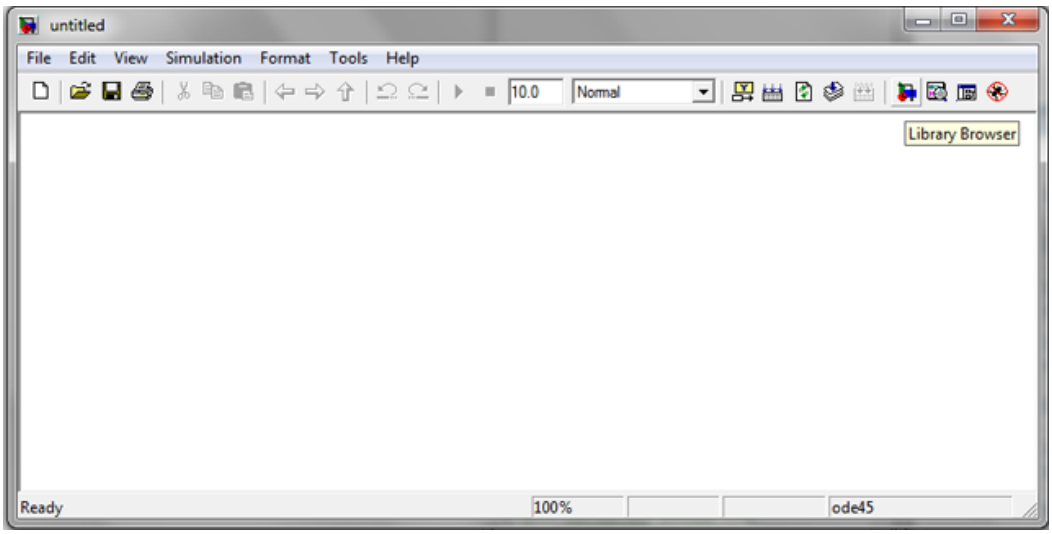
Поэтому, сразу усвоим главное правило –

MATLAB – русофобская программа. Она не переваривает кириллицу ни в каком виде. Ни в адресе, ни в окнах, ни в переменных, НИГДЕ!

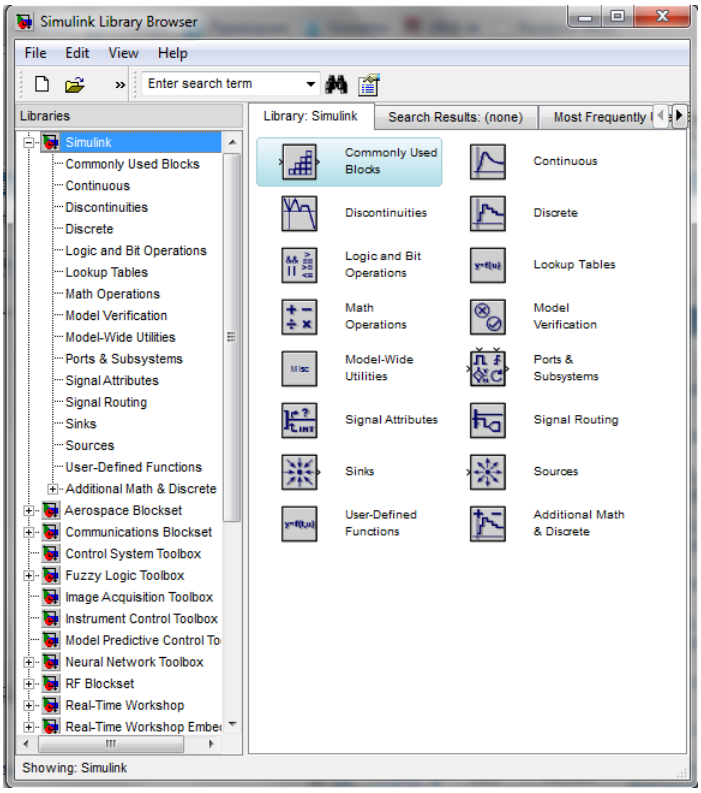
Поэтому сразу забываем о хранении наших проектов по адресу «C:\Вася\временное11\123\» и выбираем папку что-то типа «C:\Vasya\Temp11\123\».

Отлично, теперь можно начинать работать.

Нажимаем «File-New-Model». Возникает следующее окошко:



Отлично, уже что-то более дружелюбное, чем командная строка Матлаба. Интуитивно понятно, что это поле, где нужно что-то рисовать… Т.к. мышкой рисовать не получается, тыкнем на красно-сине-зелёную кнопочку «Library Browser». Откроется библиотека элементов Симулинка:

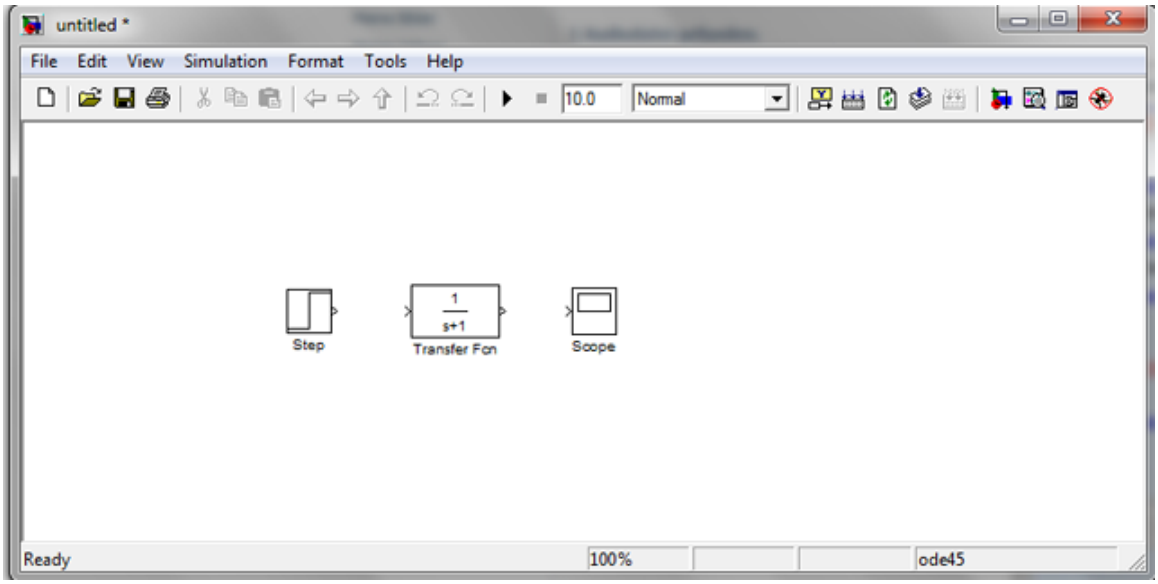


У меня до сих пор разбегаются глаза от обилия предлагаемых библиотек. Назначение очень многих из них для меня до сих пор остаются

загадкой. Мы начнём с самого простого. Соберём схему колебательного звена, посмотрим на переходный процесс при воздействии «ступеньки» и даже на ЛАХ с ЛФХ.

Для этого нам нужен источник «ступеньки», сама передаточная функция и окошко где мы будем смотреть на переходный процесс.

Предлагаю найти все эти элементы в библиотеке Simulink так, чтобы получилось что-то вроде этого:



Надеюсь, Вы не искали в библиотеке колебательное звено? В симулинке есть просто передаточная функция в которую можно вбить любые коэффициенты в знаменатель и числитель (главное, чтобы при этом порядок числителя был меньше порядка знаменателя). Но об этом немного ниже.

Сейчас постараемся соединить все эти звенья друг с другом. Можно делать это просто тягая мышкой от одной стрелочки к другой, а если надоест, можно просто выделить один блок левой клавишей, потом зажав Ctrl, нажать на другой блок.

Отлично. Теперь нужно настроить эти блоки один за другим... Про ступеньку и говорить нечего, там всё ясно, оставим всё по дефолту: время начала 1 с, начальное значение 0, конечное значение 1.

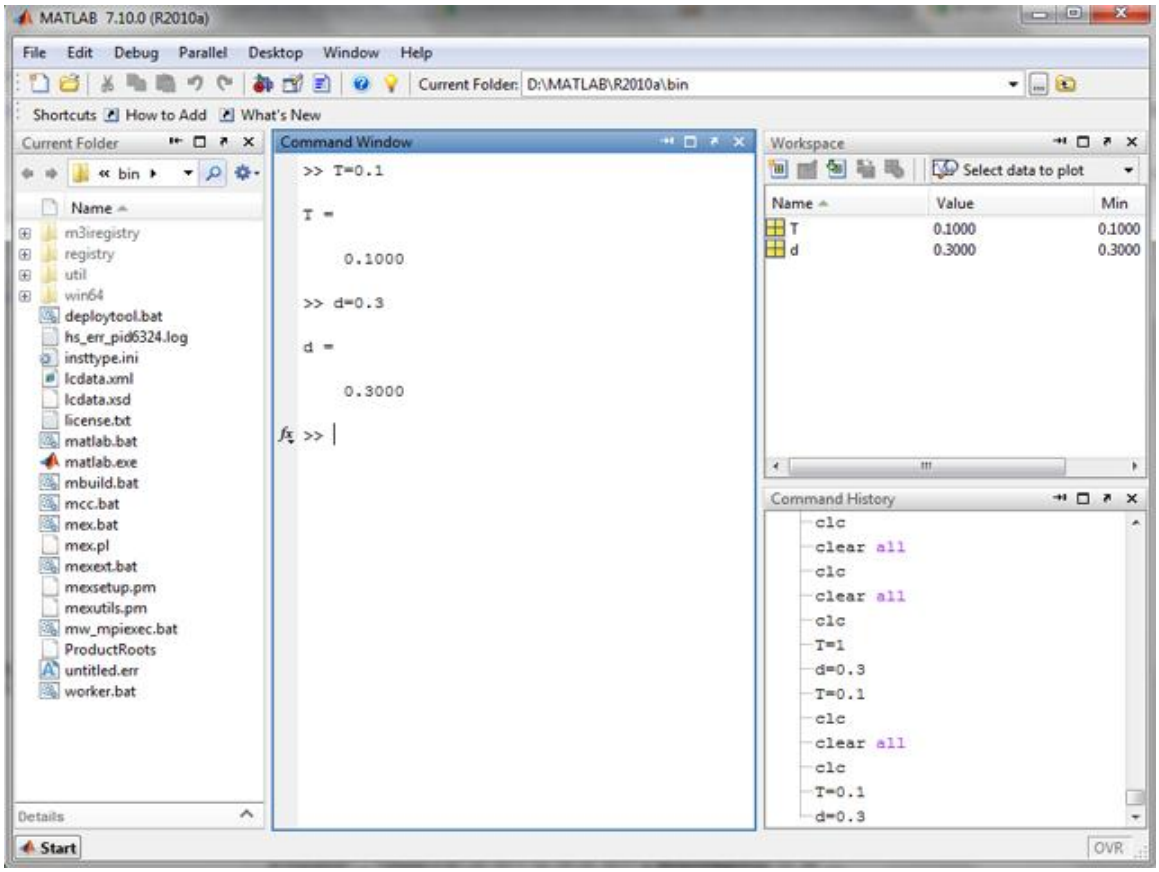
Интереснее с передаточной функцией. Напомню, что мы собирались собрать колебательное звено. Я лично привык видеть его в виде:

$$W(s) = \frac{K}{T^2 \cdot s^2 + 2\zeta T \cdot s + 1}$$

в то время как симулинк требует от нас сразу коэффициенты знаменателя и числителя. Можно, конечно, быстро перемножить на бумажке

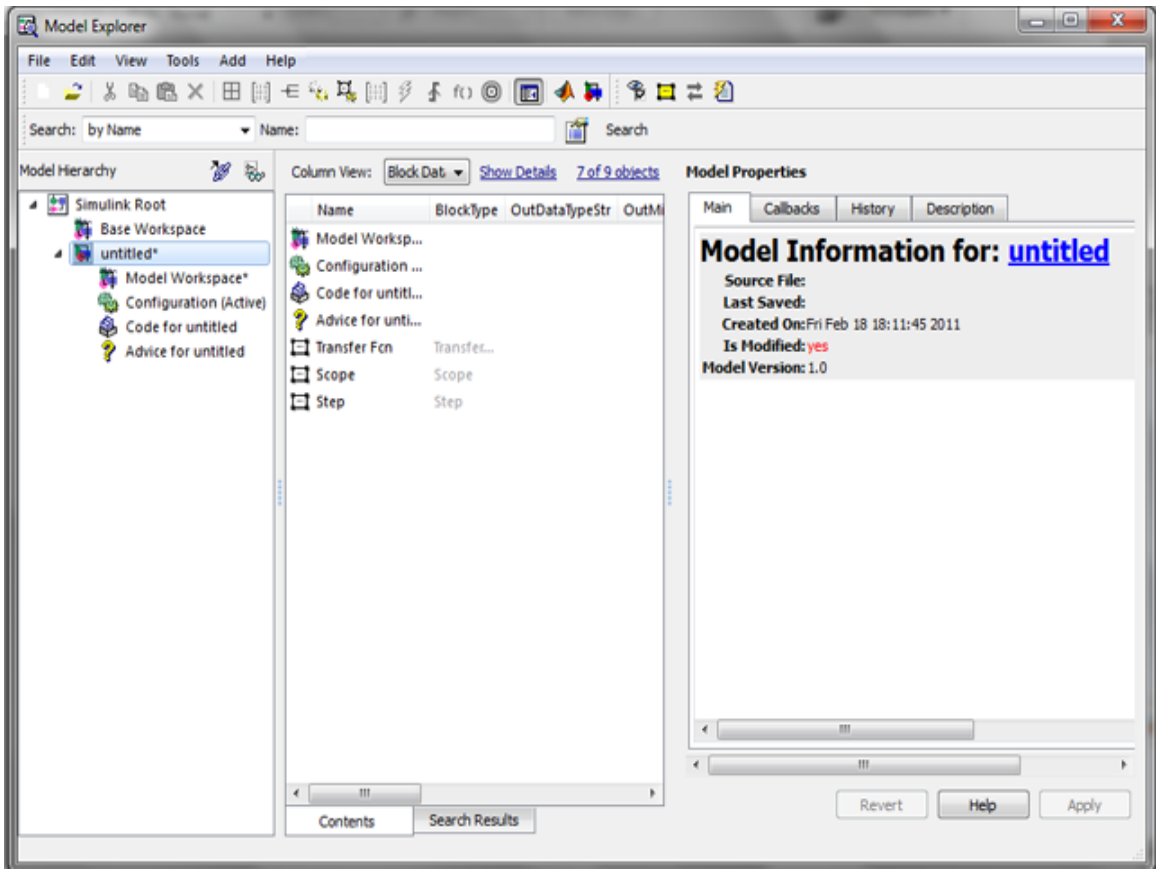
постоянную времени и коэффициент демпфирования, но это не серьёзно... Лучше научимся пользоваться переменными.

Есть как минимум 2 способа управляться с переменными в симулинке. Я привык вбивать их в командную строку Матлаба, которую мы видели в самом начале. Тогда получится что-то вроде этого:



Как видно, значения тут же внеслись в Workspace и теперь мы можем свободно вбивать их в любое окошко симулинка и проводить с ними любые операции. Потом можно сохранить их в один файл (Save Workspace as...) и загружать при необходимости.

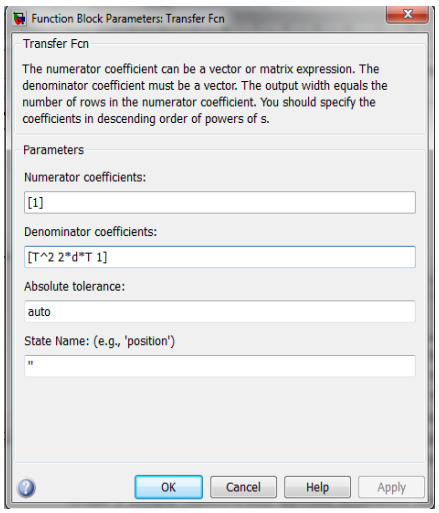
Другой вариант — делать то же самое через Model Explorer. Нажимаем на кнопку с изображением лупы рядом с кнопкой «Library Browser» и получаем следующее окно:



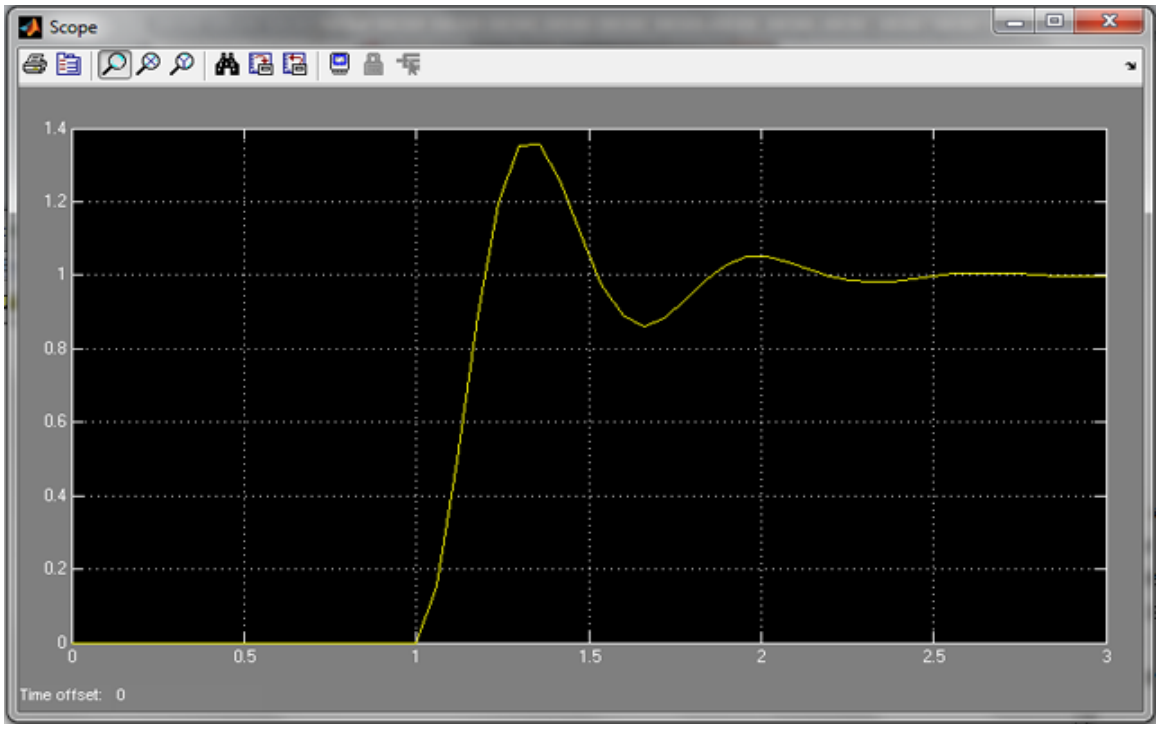
Как видно, здесь имеется не только общий Workspace, но и Workspace непосредственно модели. В общем-то, можно задать все переменные там, и тогда они будут загружаться автоматически вместе с моделью, но я всё же склоняюсь к тому, что глобальные переменные, хранящиеся в отдельном файле — это более правильно. Можете считать меня занудой ☺

Итак, вбиваем любым способом значения T=0.1 (постоянная времени колебательного звена) и d=0.3 (коэффициент демпфирования).

Теперь в окошке передаточной функции останется только перемножить всё как нужно:

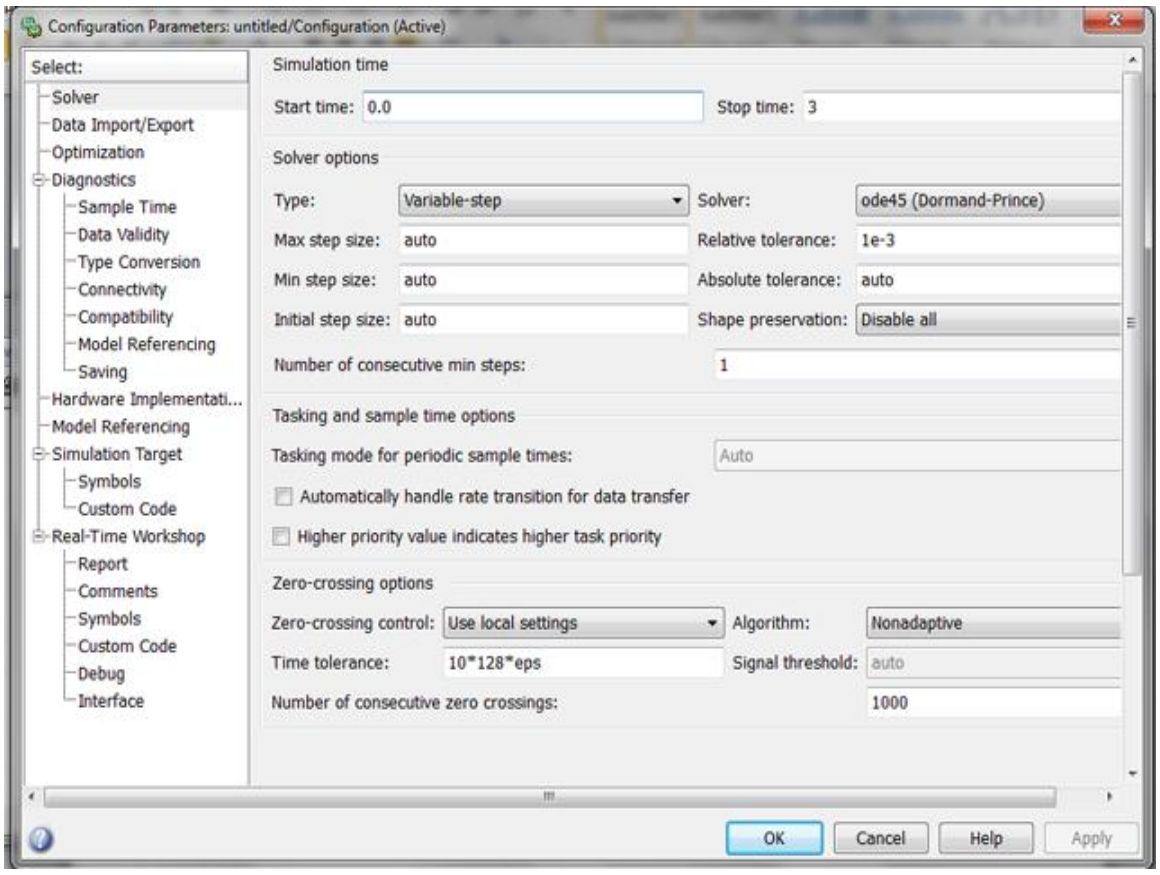


Ну всё, теперь можно запустить наконец расчёт. Жмём на «плей», затем на окошко Scope, затем на «автомасштаб» (бинокль). Самое время проанализировать результат. Получилось что-то похожее на колебательный процесс, который оканчивается примерно через 3 секунды, однако, какой-то угловатый. Ну первым делом уменьшим время моделирования с 10 до 3 секунд (справа от кнопок «плей» и «стоп») и рассмотрим по-ближе:



На лицо какие-то угловатости, особенно в области пика. Матлаб в командной строке выдал жалобу, мол «выбран дефолтный максимальный шаг столько-то, если на самом деле всё окок, можешь меня заткнуть, нажав туда-то». У нас всё не окок.

Самое время познакомиться с решателями симулинка. Нажимаем «Simulation – Configuration Parameters...»:



Опять глаза разбегаются от обилия всяких менюшек и полей. Попробуем разобраться. Сейчас нас интересует, прежде всего, решатель (solver). Здесь мы выбираем каким методом будем решать дифференциальные уравнения (да, блоки, которые мы суём в поле симулинка, на самом деле – диффуры ☺). Прежде всего, определяемся, что это будет за метод – с постоянным шагом или переменным. Есть мнение, что система диффур любой сложности решается методом Эйлера, путём уменьшения шага ☺ Это значит, что если в нашем переходном процессе длительностью 5 секунд, есть момент длительностью 0.1 с, который нужно обсчитывать с шагом 0.1 милисекунду (например, шток гидроцилиндра бьётся об упор), мы оставшиеся 4.9 секунды когда гидроцилиндр спокойно движется, должны будем обсчитывать всё равно с шагом 0.1 милисекунда. Время расчёта в этом случае увеличивается во много раз и для серьёзных задач может превратиться в несколько часов. Поэтому добрые математики придумали алгоритмы с переменным шагом. Т.е. в ответственные моменты решатель решает диффуры, уменьшая шаг настолько насколько это требуется, а когда всё «спокойно», увеличивает шаг, экономя наше время.

Бывают случаи, когда нужно решать диффуры именно с постоянным шагом, но в большинстве всё же переменный шаг значительно лучше.

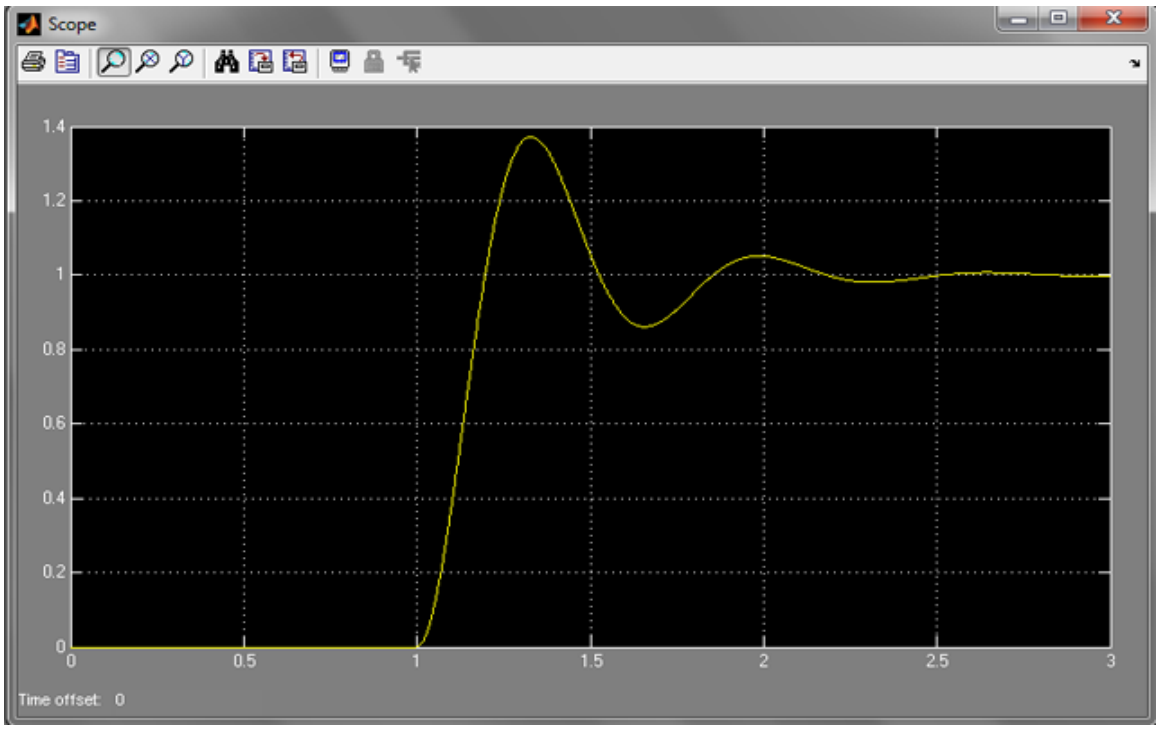
Следующее меню – сам решатель. Это вообще - глубокий матан, вникать в него особо не будем, пройдёмся вкратце. Хэлп Матлаба очень

уж рекомендует решатель ode45. Из приведённых там таблиц следует, что он самый точный. С одной лишь оговоркой. Если наша задача не жёсткая. Под жёсткостью понимается не содержание неимоверной жести в условии, а жёсткость или нежёсткость диффур нашей мат. модели. А между ними в свою очередь грань довольно расплывчатая. Один и тот же диффур может быть как жёстким, так и не жёстким, в зависимости от коэффициентов. В нашем случае — это постоянные времени различных элементов системы. Если они различаются на несколько порядков (как например, постоянная времени хорошего ЭГУ и плохого гидроцилиндра), то скорее всего нежёсткий решатель типа ode45 пойдёт в разнос и выдаст нам мегарасходящийся переходный процесс. В этом случае нужно использовать жёсткие решатели, помеченные буквой s (stiff). Тут наиболее точным является решатель ode15s.

Резюмируя. ***Тривиальные задачи решаем с ode45, задачи с элементами системы с заведомо различающимися постоянными времени (то как наши родные гидроприводы), решаем с ode15s.*** *Если ничего не получается, ставим метод Рунге-Кутты 4-го порядка (ode4) с шагом 0.1 мкс и оставляем комп считать на ночь ☺*

Поехали дальше. А дальше всё значительно проще… Нужно всего лишь выбрать максимальный, минимальный и начальные шаги. Вспоминаем, что матлаб жаловался на максимальный шаг, ставим значение 0.001 с для перестраховки. Теперь, даже если решатель вздумает, что ничего сложного на его пути нет, он всё равно будет считать отрезками не больше 1 мс.

Жмём ок и затем «плэй». Смотрим результат:



Другое дело. Всё гладенько, да и матлаб больше не жалуется. Можно хоть сейчас вставлять в чьё-нибудь ДЗ ☺

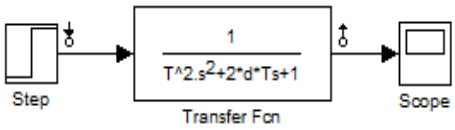
Ну кстати, сразу пару слов о графиках. Рядом с кнопкой принтера, есть кнопка «Парметры» в которой можно настроить, к примеру, количество осей. В этом случае у окошка Scope появится несколько входов, и графики будут располагаться один под другим. Чтобы расположить несколько графиков на одном, нужно использовать блок Mux

В этом случае несколько сигналов можно запихнуть в один Scope и в окне они будут отображаться разными цветами.

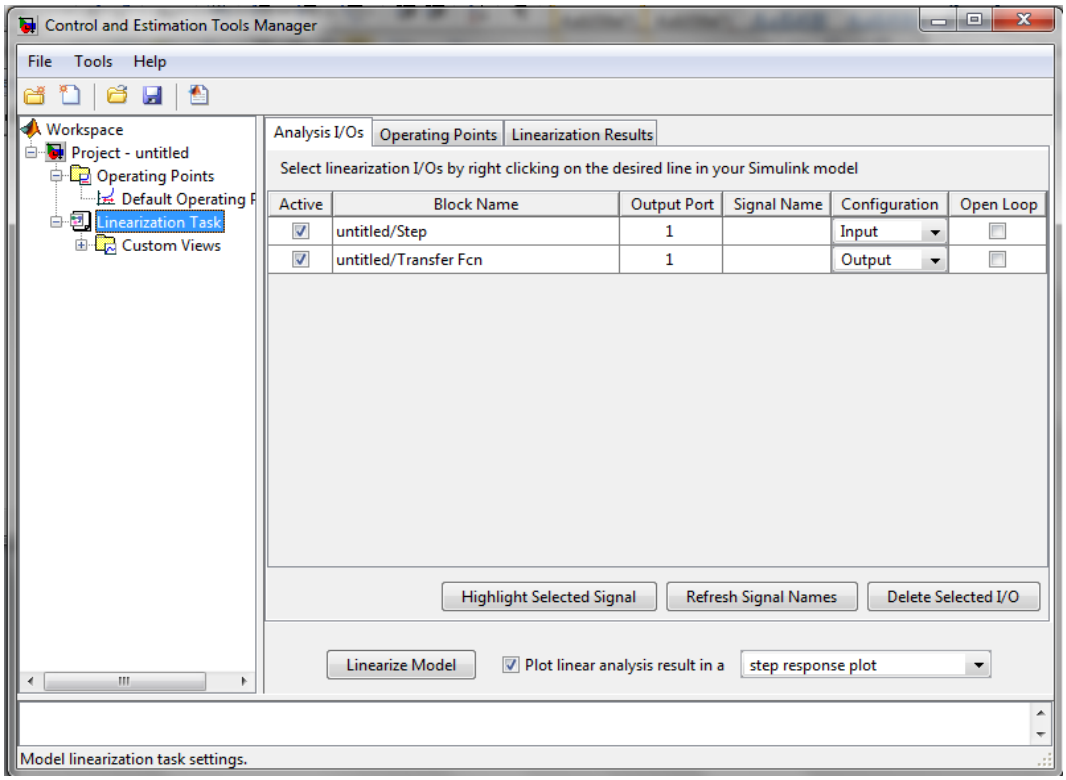
Control Design

Теперь разберёмся как строить логарифмические амплитудные и фазовые характеристики. Делается это при помощи расширения «Control Design» там же в симулинке.

Для начала нужно показать системе где у нас вход и где выход. Жмём правой кнопкой на линию, между ступенькой и передаточной функцией, выбираем Liniarization Points – Input Point. Между передаточной функцией и выходом, соответственно ставить Output Point. Теперь всё это должно выглядеть примерно так:



Теперь в выпадающем меню Tolls жмём Control Design – Linear Analysis:

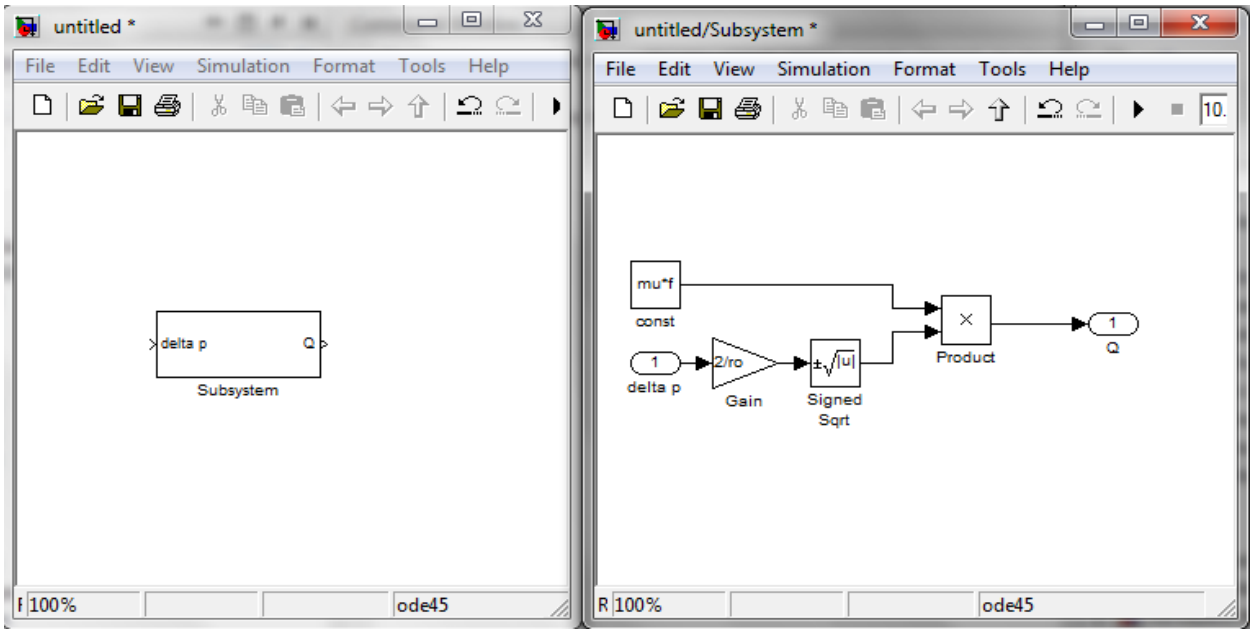


Тут уже обозначены точки, которые мы расставили. При желании тут же можно поменять выходы на входы, если хочется, или вообще деактивировать некоторые точки. В общем, долго думать тут не над чем, жмём Linearize Model.

Программа долго думала и выдала нам переходный процесс, который мы, собственно, уже видели. Но это не всё. Жмём на правую кнопку на поле графика, Plot Types – Bode. И о чудо – мы видим знакомые с детства ЛАХ и ЛФХ колебательного звена. Поигравшись в этом окошке, можно не только нарисовать логарифмическую сетку и изменить размерность горизонтальной оси с рад/с на Герцы, но и в автоматическом режиме расставить запасы по фазе и амплитуде; а также получить очень полезное замечание Капитана Очевидность о том, что система устойчива (если запас есть) или неустойчива (если запаса нет). Одним словом – мечта.

Simscape

Когда я более-менее освоился с симулинком и стал моделировать гидроприводы, меня посетила мысль, что круто было бы создать библиотеку стандартных гидравлических элементов, которые я постоянно использую в моделях (то как дроссель, гидроцилиндр, кромки распределителя). Благо, в симулинке при помощи блока Subsystem это делается довольно просто. Вот так, например, выглядит модель дросселя, где на входе задаётся перепад давлений, а на выходе получается расход:

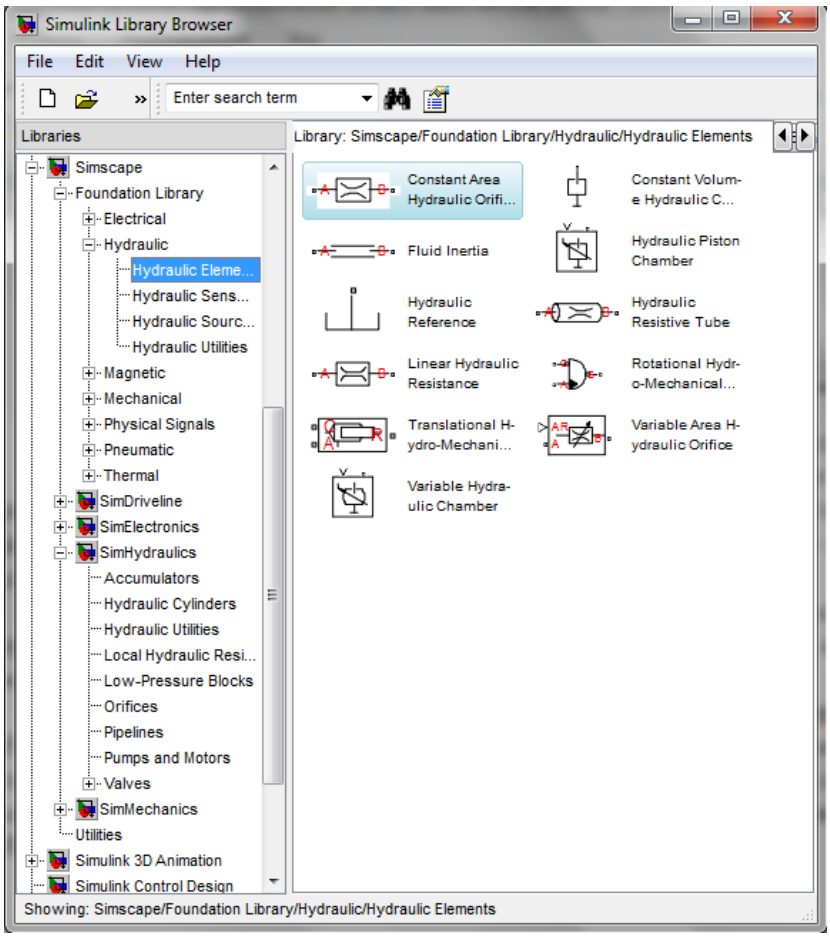


Теперь этот блок можно вставлять во все схемы, где нужен дроссель...

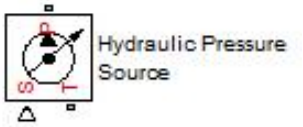
Собственно, я, наверное, так и клепал бы эти блоки, если бы не стал рыться в библиотеках симулинка и не наткнулся на ***simscape****, где вся эта работа уже была проделана, а самое главное — содержание каждого блока известно и подробно описано со всеми формулами в хелпе.*

Для того, чтобы быстро освоиться с этой библиотекой, соберём простенькую схему, состоящую из гидроцилиндра, распределителя 4/3 и какого-то источника постоянного давления.

Для этого, найдём в огромном списке библиотек, библиотеку ***Simscape****. В ней имеется куча разных библиотек для моделирования механических, электрических, магнитных и других физических процессов. Нас сейчас интересует, прежде всего, гидравлические элементы, а они находятся в библиотеке* ***Foundation Library/Hydrailic*** *и* ***SimHydraulics****.*

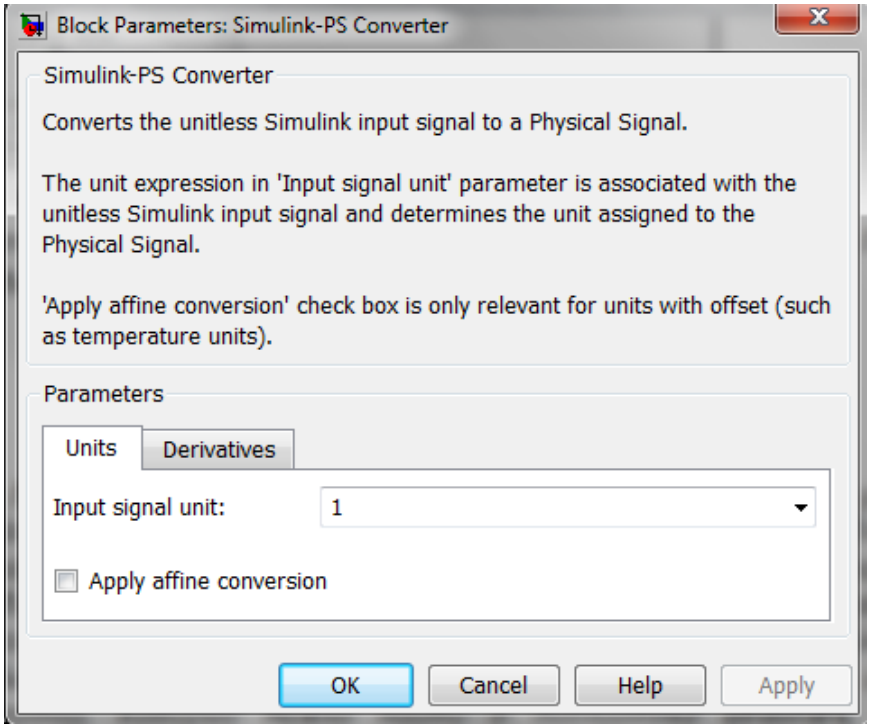


Собственно, теперь осталось только перетащить необходимые элементы на форму. Проще всего найти источник давления в библиотеке основных элементов:



Тут всё просто. 2 «гидравлических» входа и 1 вход для управляющего сигнала, который, собственно и определяет, какое давление будет отправлено к линии P, относительно линии T. Однако, вход для управляющего сигнала не «простой». На него нельзя отправить просто сигнал из блоков симулинка, т.к. этот сигнал «физический». Для того, чтобы передать этому блоку какое-то значение, нужно использовать конвертер, который можно найти в библиотеке ***Utilities****, а именно –* ***Simulink-PS Converter****.*

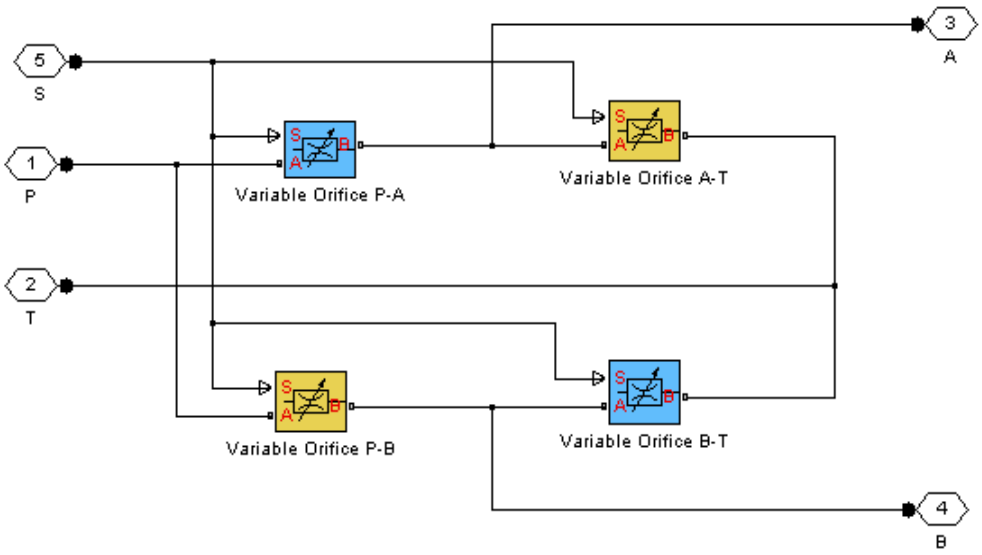
На его вход можно отправлять сигналы симулинка, а он будет их преобразовывать в сигналы для Simscape. Но это ещё не всё. Откроем его параметры:



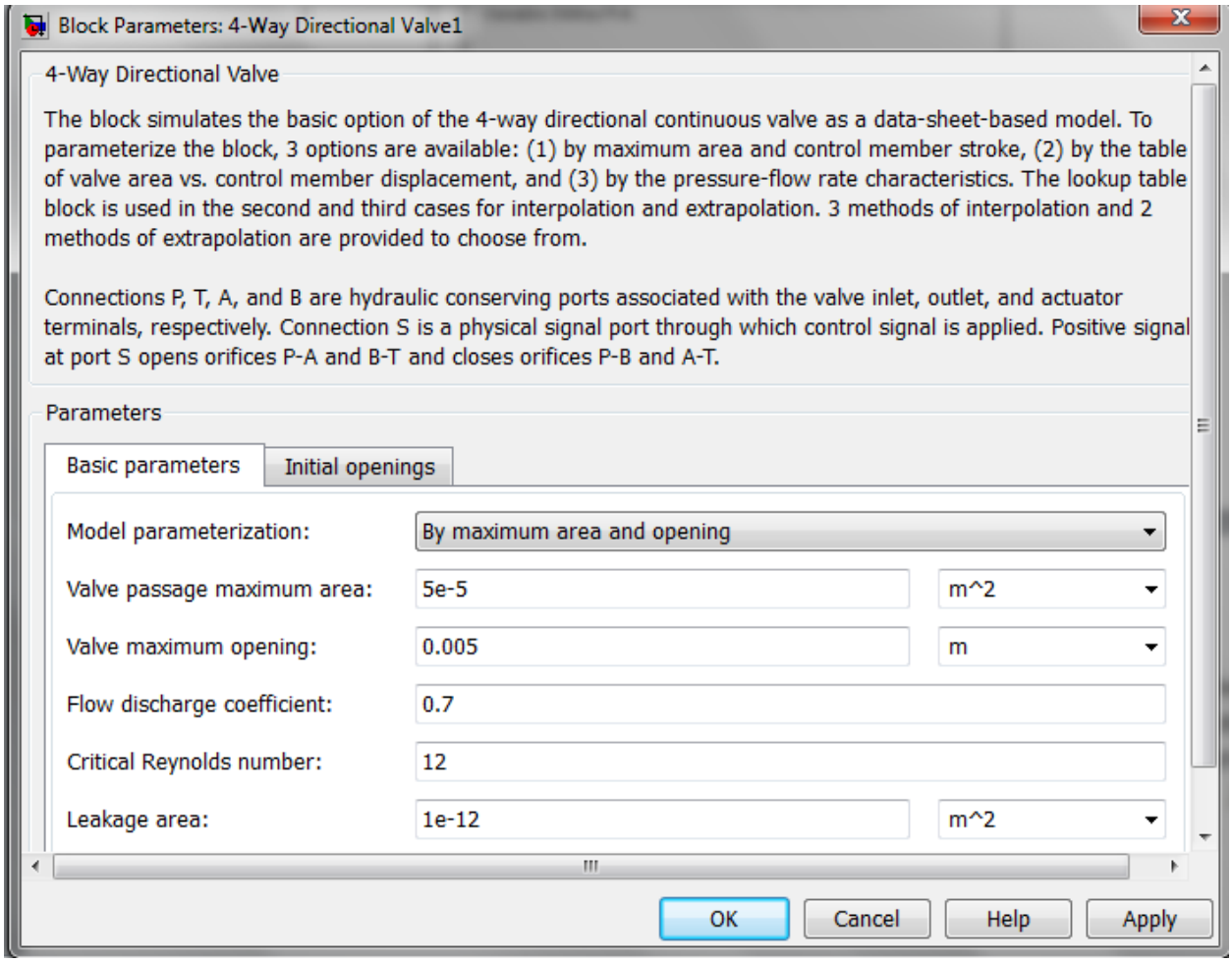
В этом окне можно изменять размерность сигнала. Т.е. если мы не хотим отправлять на вход константу в виде «16e6», чтобы заставить блок давления выдавать постоянно 16 МПа, можно просто в этом окошке вписать «MPa» (без кавычек) или, ещё лучше, «bar». Очень удобно.

С источником давления теперь точно всё ясно, перейдём к распределителю. Если заглянуть в раздел ***Valves/Directional Valves****, глаза разбегаются...*
Нам нужен простенький распределитель 4/3 с закрытым центром.
Выбираем ***4-Way Directional Valve****:*

Вроде всё ясно, 4 гидравлических канала (давление, слив, 2 линии A и B) и вход для управляющего сигнала, который отвечает за смещение золотника. Посмотрим, что у этого блока внутри по версии хэлпа:



У нас есть 4 кромки, которые меняют своё сопротивление в зависимости от входного сигнала. Всё как в жизни ☺ Теперь осталось только определиться как будут меняться сопротивления этих щелей в зависимости от открытия. Это выбирается в окне свойств этого распределителя:



Есть 3 варианта задания статических характеристик распределителя:

- *по максимальному открытию и соответствующей ему площади (в этом случае расход при постоянном перепаде будет изменяться линейно)*
- *по таблице соответствия смещения золотника и площади открытия*
- *по таблице расходно-перепадной характеристики.*

Мы не будем в этом учебном примере особо заморачиваться с расходно-перепадной характеристикой, выберем самый простой вариант по максимальному открытию. Введём следующие параметры:

Максимальное смещение золотника — 3мм

Максимальная площадь открытия — 1.855e-06 $м^2$

Коэффициент расхода — 0.63

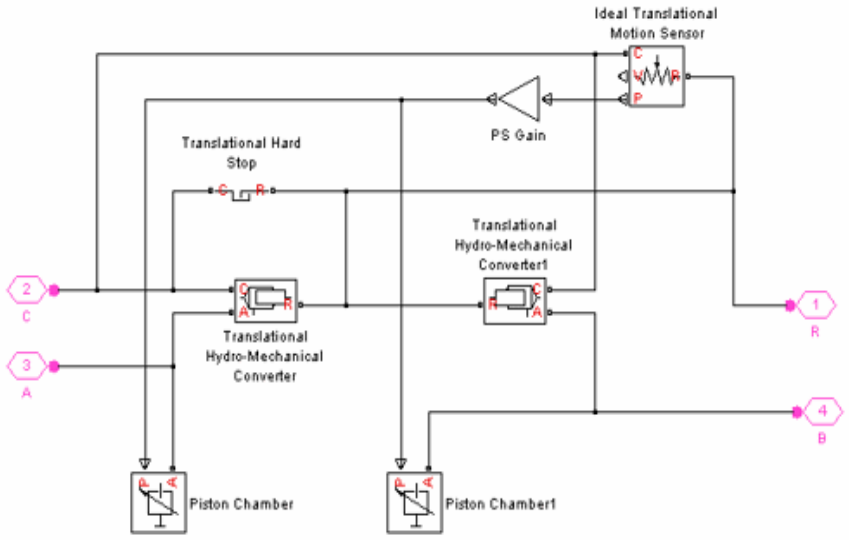
Число Рейнольдса после которого режим становится ламинарным — 12 (по дефолту)

Площадь «дырки» в золотнике в нуле — тоже по дефолту.

Теперь самое интересное - гидроцилиндр. Условимся, что это будет гидроцилиндр с непроходным штоком. В модели будем учитывать сжимаемость жидкости, а также удар об упор.

*В разделе **Hydraulic Cylinders** как всегда куча непонятных элементов. Из них в моей версии симулинка видно как минимум 4 варианта гидроцилиндров, половина из которых одностороннего действия. Остаётся выбрать из 2-х гидроцилиндров. Для этого придётся заглянуть в хэлп и проглядеть по диагонали описание каждой модели. Если описать это в двух словах, то упрощённая модель гидроцилиндра (**Double-Acting Hydraulic Cylinder (Simple)**) не учитывает сжимаемость жидкости. Это значит, что скорость гидроцилиндра будет прямо пропорциональна поступающему расходу, а давление — внешней нагрузке. Это совсем не интересно...*

*Рассмотрим лучше более полную модель гидроцилиндра **Double-Acting Hydraulic Cylinder**. Схематично она выглядит так:*



*Основные элементы – 2 поршня (**Translational Hydro-Mechanical Converter**), соединённые одной механической связью, которые просто делят поступающий в них (или выходящий из них) расход на площадь (которую мы задаём) и выдают скорость перемещения на механическую связь. Т.е. без остальных прибамбасов это тот же упрощённый гидроцилиндр. Однако, к полостям этих поршней подсоединены изменяемые объёмы **Piston Chamber**. Объём в этом блоке вычисляется как площадь поршня, умноженная на позицию гидроцилиндра в данный момент.*

Про этот блок тоже лучше почитать в хэлпе, но общий смысл в том, что давление в этом случае изменяется не внезапно, а по следующей зависимости:

$$\dot{p} = \frac{E}{V} \cdot \sum Q$$

где

$\dot{p}$ *– производная давления по времени (скорость изменения давления)*

E *– модуль упругости жидкости (который зависит от давления. Эта зависимость обсчитывается в этом блоке, и формула написана в хэлпе)*

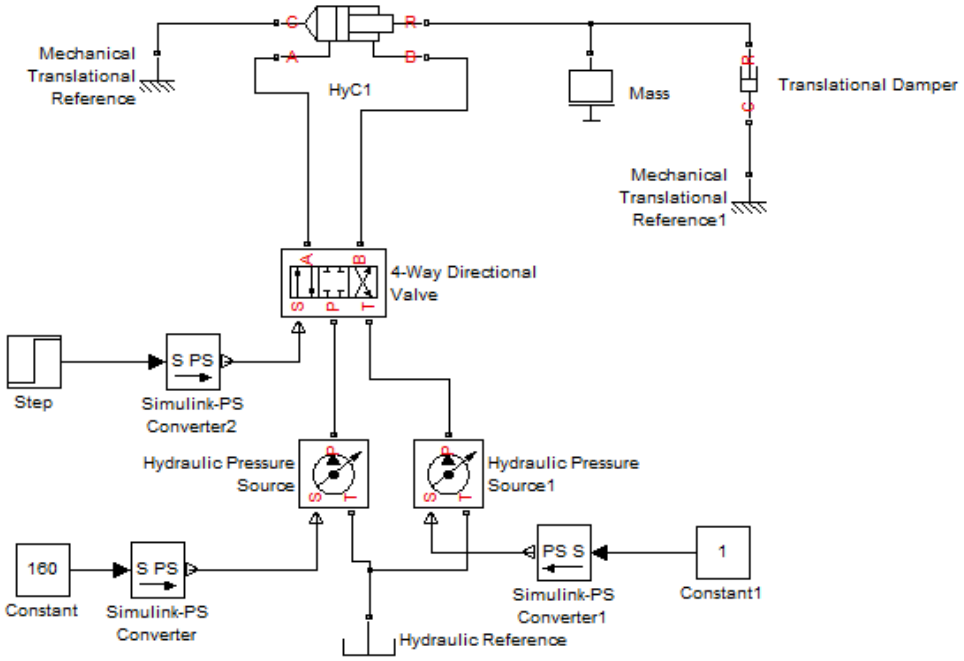
V *– объём жидкости*

$\sum Q$ *– алгебраическая сумма расходов, поступающих в этот объём (расход поступающий в объём, увеличивает давление, расход выходящий из объёма его уменьшает. Всё логично ☺)*

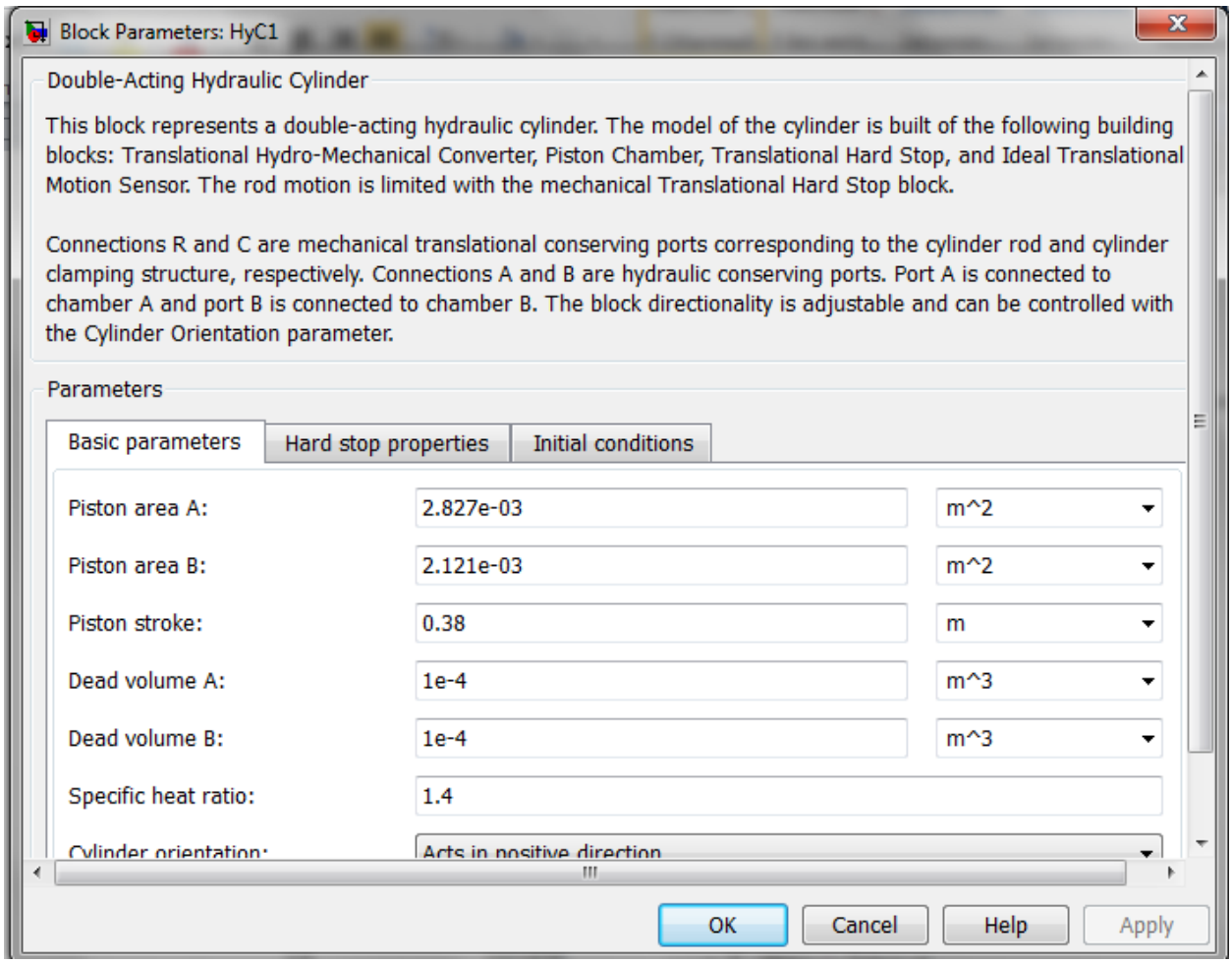
Ну и помимо этого в модель добавлен жёсткий стоп, который представляет из себя пружину и демпфер, которые начинают работать, когда поршень доходит до одного из крайних положений.

Как-то неожиданно в этой модели появился новый вид канала – механический. Логично, что гидроцилиндр для своей продуктивной работы должен быть, во-первых, как-то закреплён, а во-вторых, к чему-то подсоединён. Пусть в нашем случае он будет закреплён абсолютно жёстко и будет двигать какую-то массу.

Все необходимые элементы можно найти в разделе ***Foundation Library/Mechanical****. Опору C гидроцилиндра соединим с «землёй», а к штоку прицепим массу и демпфер. В итоге в сборе схема будет выглядеть примерно так:*



Параметры гидроцилиндра списываем отсюда (они нам понадобятся далее):



Параметры жёсткого стопа и начального положения оставляем по дефолту.

Масса — 2000 кг.

Параметры демпфера по умолчанию.

Ступенька высотой 3 (конвертер настроен на миллиметры) через время 0.1 с.

Вроде всё... Но если запустить моделирование, то ничего хорошего не произойдёт. Вылезет куча ошибок, да и пронаблюдать мы ничего не сможем.

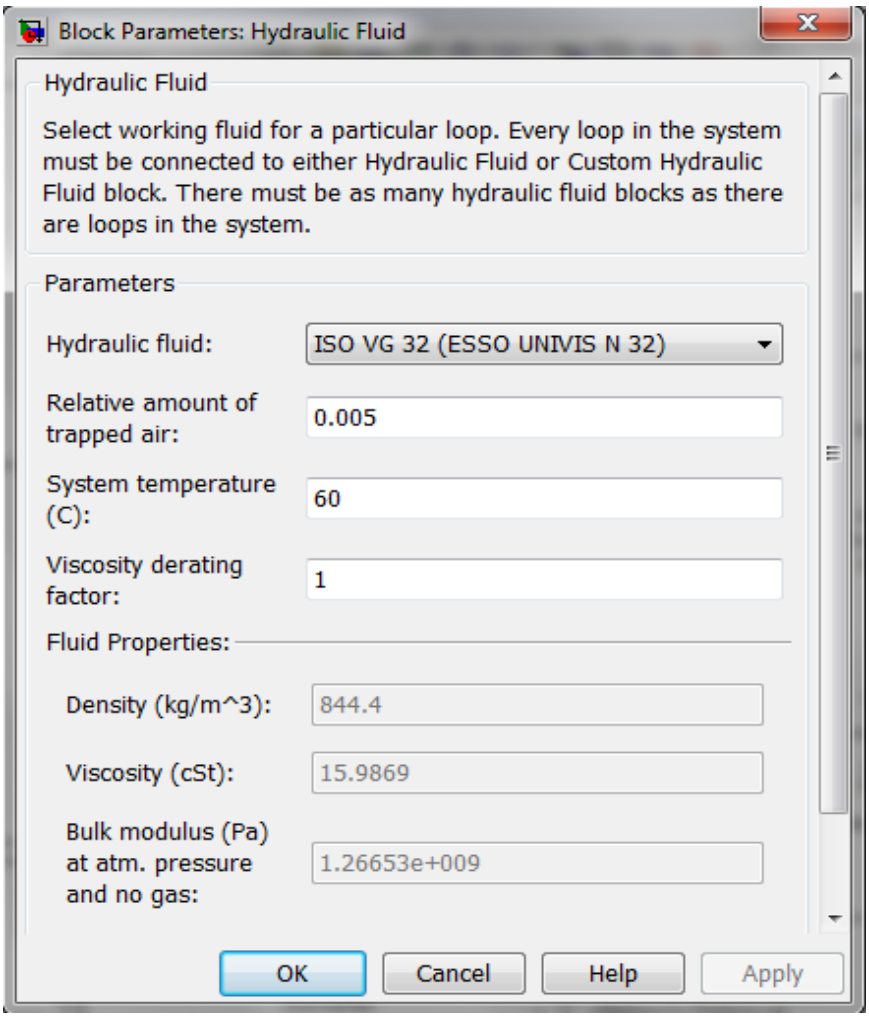
Не хватает 3-х важных вещей:

Во-первых, блока с параметрами жидкости.

Во-вторых, окошек где мы будем наблюдать изменение параметров.

Ну и в-третьих — блока решателя, который жизненно необходим всем моделям симскэйпа.

Начнём с самого простого — подсоединим к нашей системе в любом месте блок жидкости ***Hydraulic Fluid*** *и сразу залезем в его параметры:*

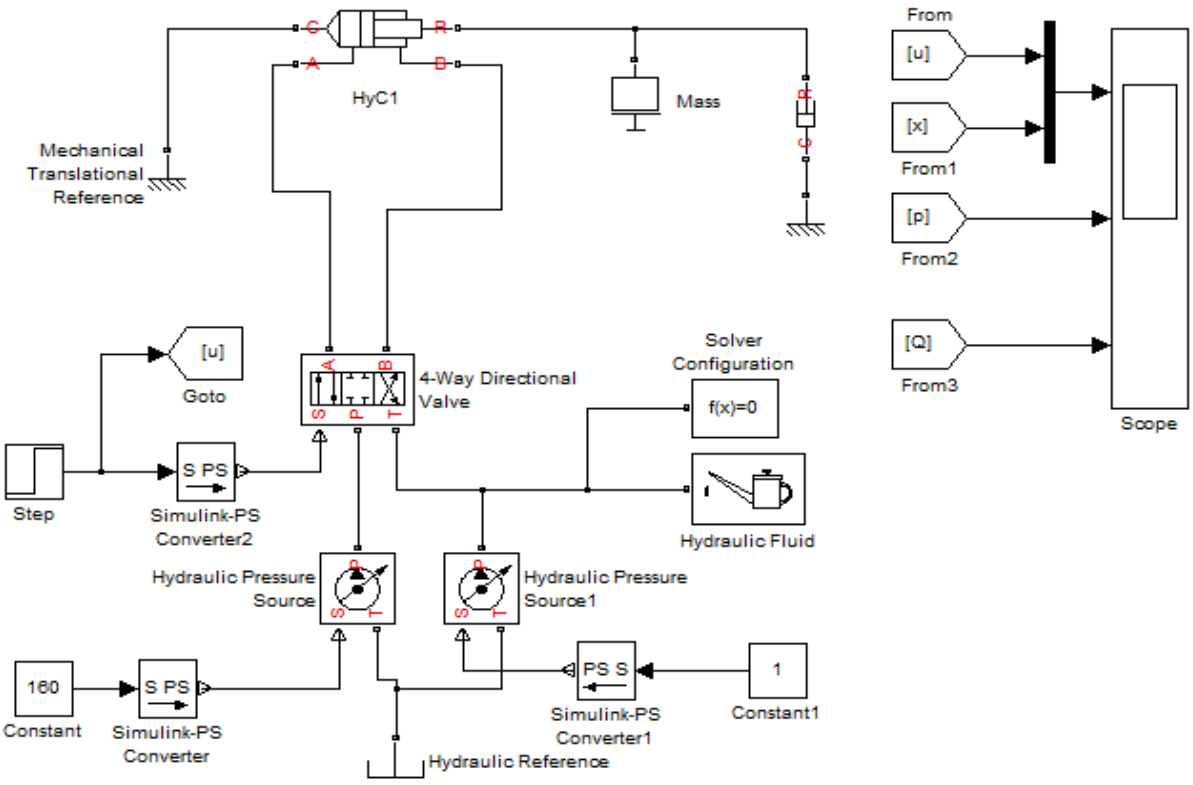


Тут можно выбрать из списка уже имеющиеся жидкости и в нашей системе тут же пропишутся параметры плотности, вязкости, содержания воздуха, модуль упругости и т.д... Мне понравилась жидкость, как на скриншоте, её я и выбрал ☺

*Сразу же подключим и решатель. Его можно найти в разделе **Utilities**, и изменять в нём ничего первое время особой нужды нет.*

Гораздо интереснее дело обстоит с измерениями. Постараемся вывести на один график сигнал с золотника и перемещение штока гидроцилиндра. Помимо этого интересно посмотреть, как меняется расход через золотник и давление в рабочих полостях.

*Проще всего вывести на график управляющий сигнал. Т.к. наша схема уже довольно нагромождена, лучше сделать это, используя блоки **Goto** и **From** в разделе библиотеки симулинка **Signal Routing**. Выглядеть это будет примерно так:*



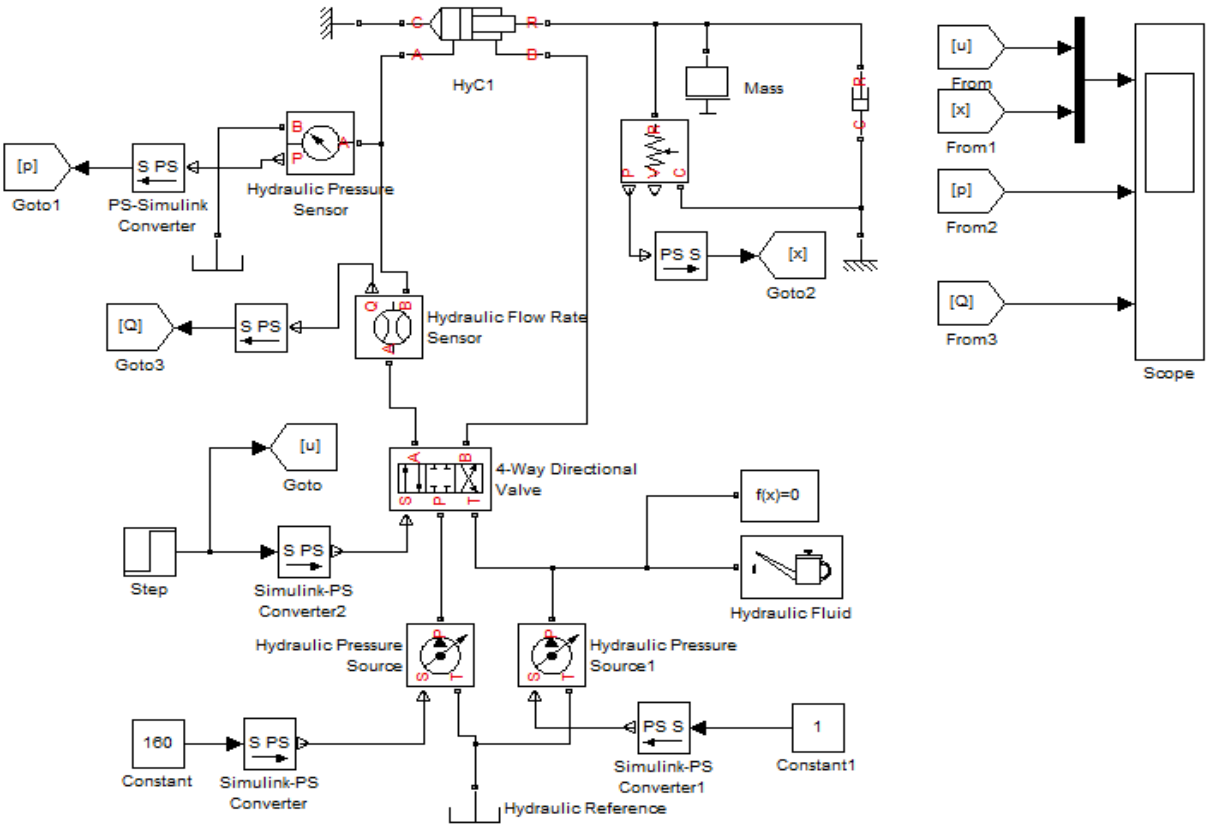
Как видно, из желаемых 4-х сигналов, мы «взяли» через Goto только один… Разберёмся как добыть остальные.

*Проще всего (как и в жизни) дело обстоит с показанием давления. Для этого мы просто берём датчик давления (**Hydraulic Pressure Sensor**), подсоединяем его к нужной нам точке, другим концом опускаем в «ведро» (либо к другой полости гидроцилиндра, если хотим мерить перепад) и сигнал отправляем через конвертер (настроенный на бары или МПа) в **Goto**.*

Аналогично дело обстоит с датчиком перемещения, который аналогично одним концом присоединяем к штоку, а другим к «земле». Конвертер настраиваем на метры.

Датчик расхода же нужно встраивать прямо в линию так, чтобы весь расход шёл через него. Если подсоединить его к «ведру», то получится, что весь расход просто уйдёт на слив и в гидроцилиндр не пойдёт ☺ Конвертер настраиваем на l/min.

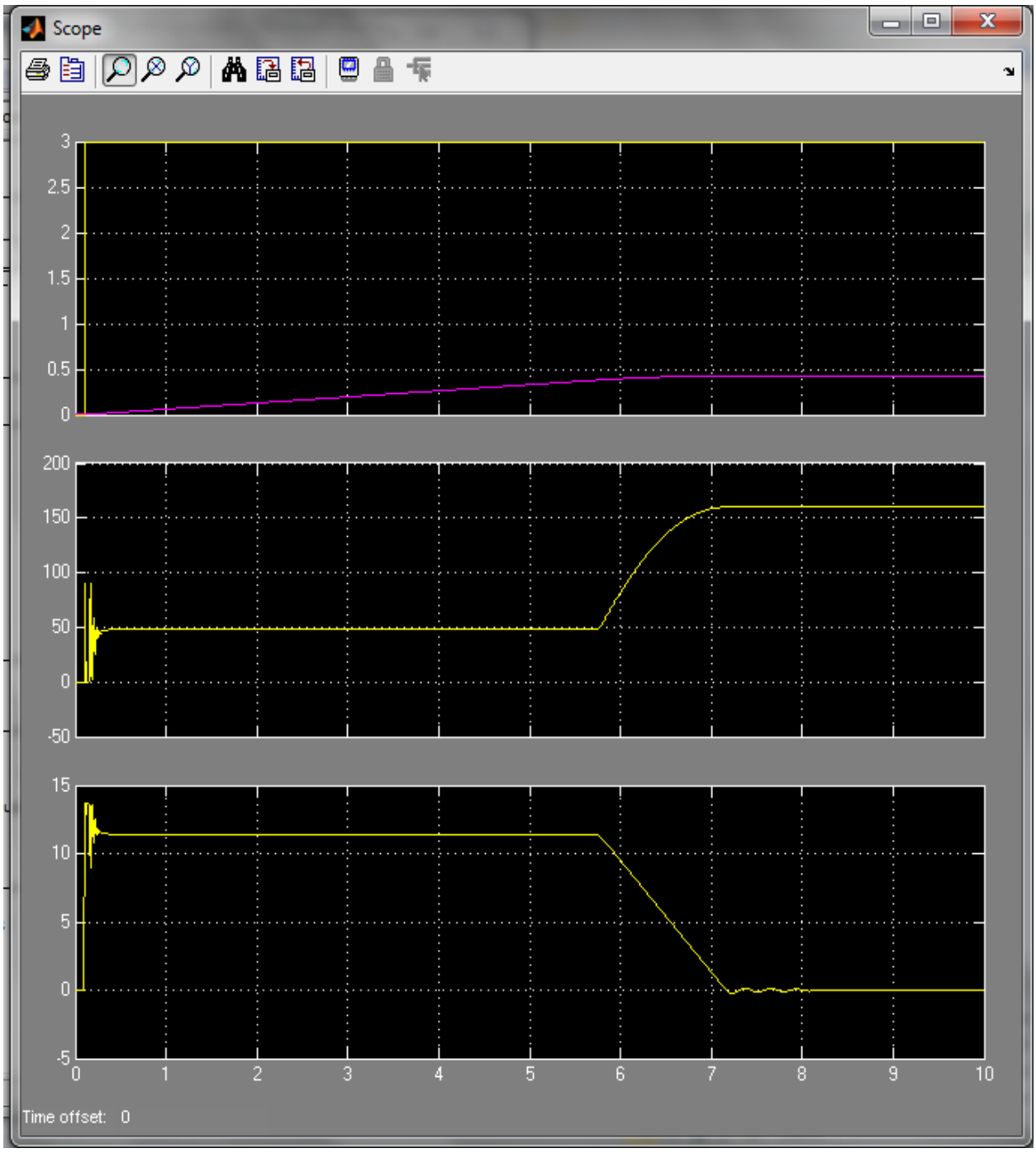
В итоге, схема выглядит примерно так:



Ахтунг!

Настоятельно рекомендуется перед запуском, уменьшить максимальный шаг интегрирования до 0.001 с, и в параметрах графика снять ограничение на максимальное количество выводимых точек. В противном случае получите только кусок графика.

После запуска у меня получилась примерно следующая картина:



Сразу видно, что явно неудачно выбран масштаб вывода управляющего сигнала и перемещения гидроцилиндра. Это можно исправить, если перед отправкой на график, разделить сигнал на какое-то число, или вообще по всем величинам перейти к безразмерному виду. Но это уже «шлифовка».

Уже сейчас можно видеть, что модель реагирует более-менее адекватно. При открытии золотника давление и расход скачком возрастают, и шток начинает двигаться. При наезде на жёсткий стоп (который, как видно, оказался простой пружиной), давление возрастает до максимального значения, а расход соответственно уменьшается до нуля. Поршень останавливается.

Собственно, таким образом, мы создали модель гидропривода, в которой волей-неволей учли столько мелочей, которые при составлении обычных диффур и не подумали бы учитывать (изменение модуля упругости жидкости от давления, жёсткий стоп и т.п.).

Ну вот и всё. На этом вводная часть закончена, и я было хотел начинать самое интересное – главу про моделирование сложных механизмов в ***SimMechanics*** *и визуализация их работы, ради чего, собственно, и была затеяна идея с методичкой, и ради чего я написал визуализированную мат. модель манипулятора, управляемую от джойстика… Однако, во-первых, на это не хватило времени из-за диплома, а во-вторых, это по большому счёту никому и не нужно ☺*

*В любом случае, надеюсь, то, что получилось, будет кому-нибудь полезно. Если появятся вопросы (которые не удастся разрешить как минимум за 2-3 бессонные ночи возни с хелпом матлаба), обращайтесь через вконтакт (*http://vkontakte.ru/max.andreev*) или в оффлайне (на момент написания статьи – комната А1 в лаборатории Э10) ☺*

Старший оберлаборант

Макс Андреев

P.S. Перечитать всю методичку полностью мне так и не удалось, так что, либо не обращайте внимания на многочисленные очепятки и бессмысленные повторения местоимений и бессмысленно повторяющихся предлогов, либо ждите второй редакции ;)