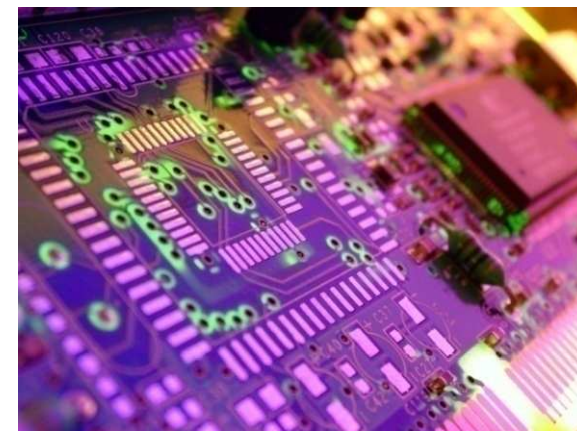


Новые возможности в модельно - ориентированном проектировании с использованием MATLAB и Simulink

Аркадий Туревский

Темы доклада

- ПЛИС и встраиваемые системы
- Совместное моделирование различных физических доменов
- Упрощенная разработка систем управления
- Генерация С кода из MATLAB



ПЛИС применяются повсюду



Сколько проектов на ПЛИС содержат алгоритмы цифровой обработки сигналов?

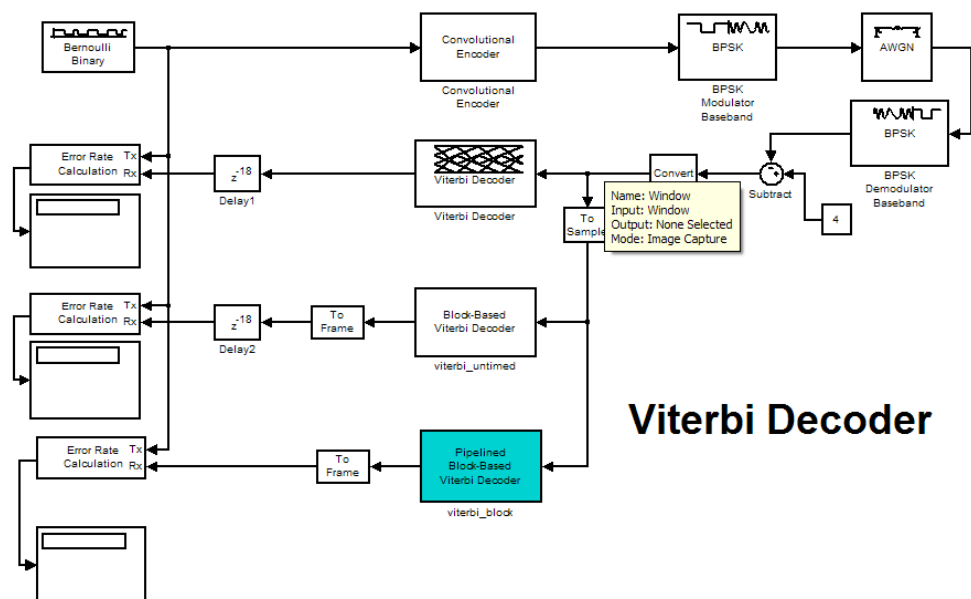
**2010 год:
начато 80,000 проектов**

**25–30% проектов
содержат алгоритмы
цифровой
обработки
сигналов на ПЛИС**

Основано на данных Gartner's

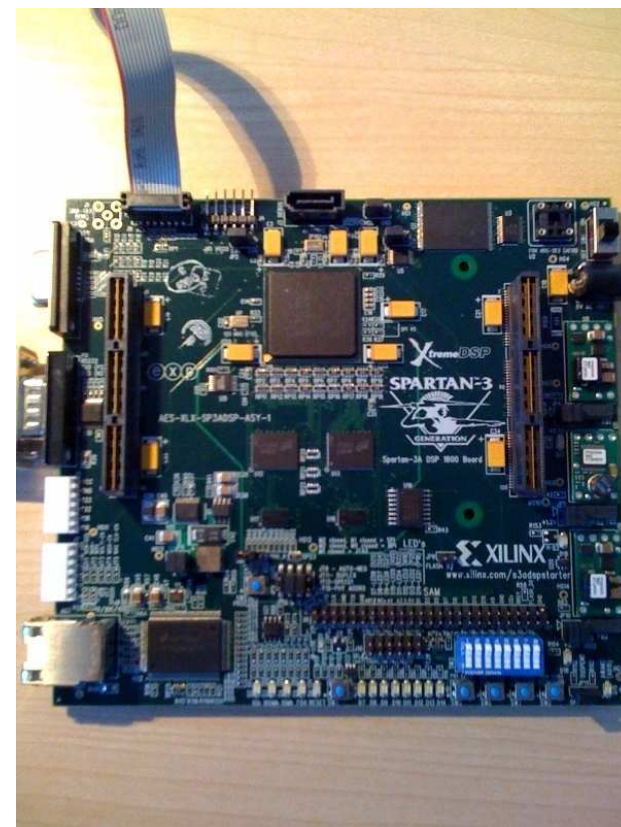
Алгоритмы и их реализация в железе

Разработчик ПЛИС



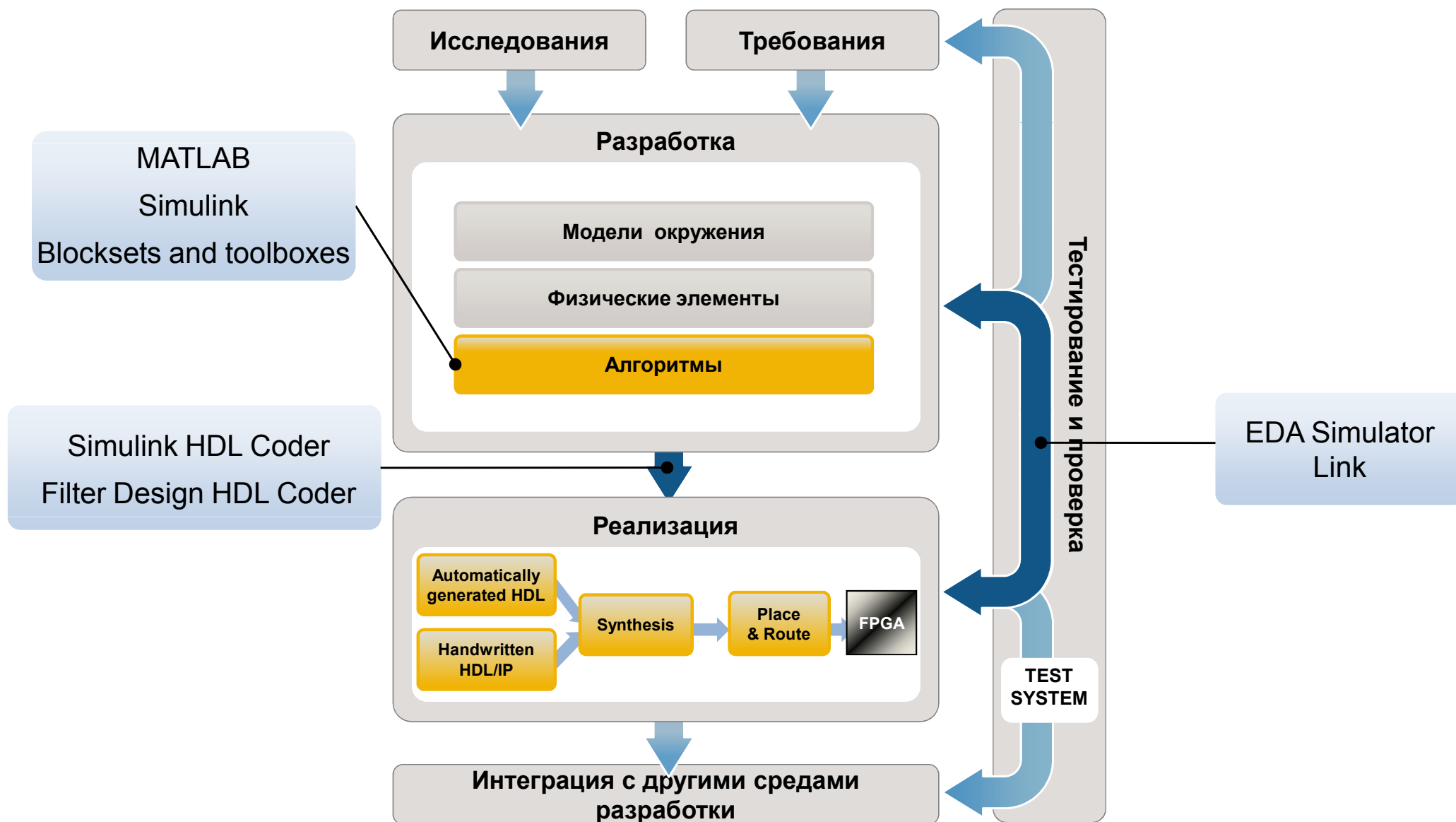
Viterbi Decoder

Разработчик
алгоритмов

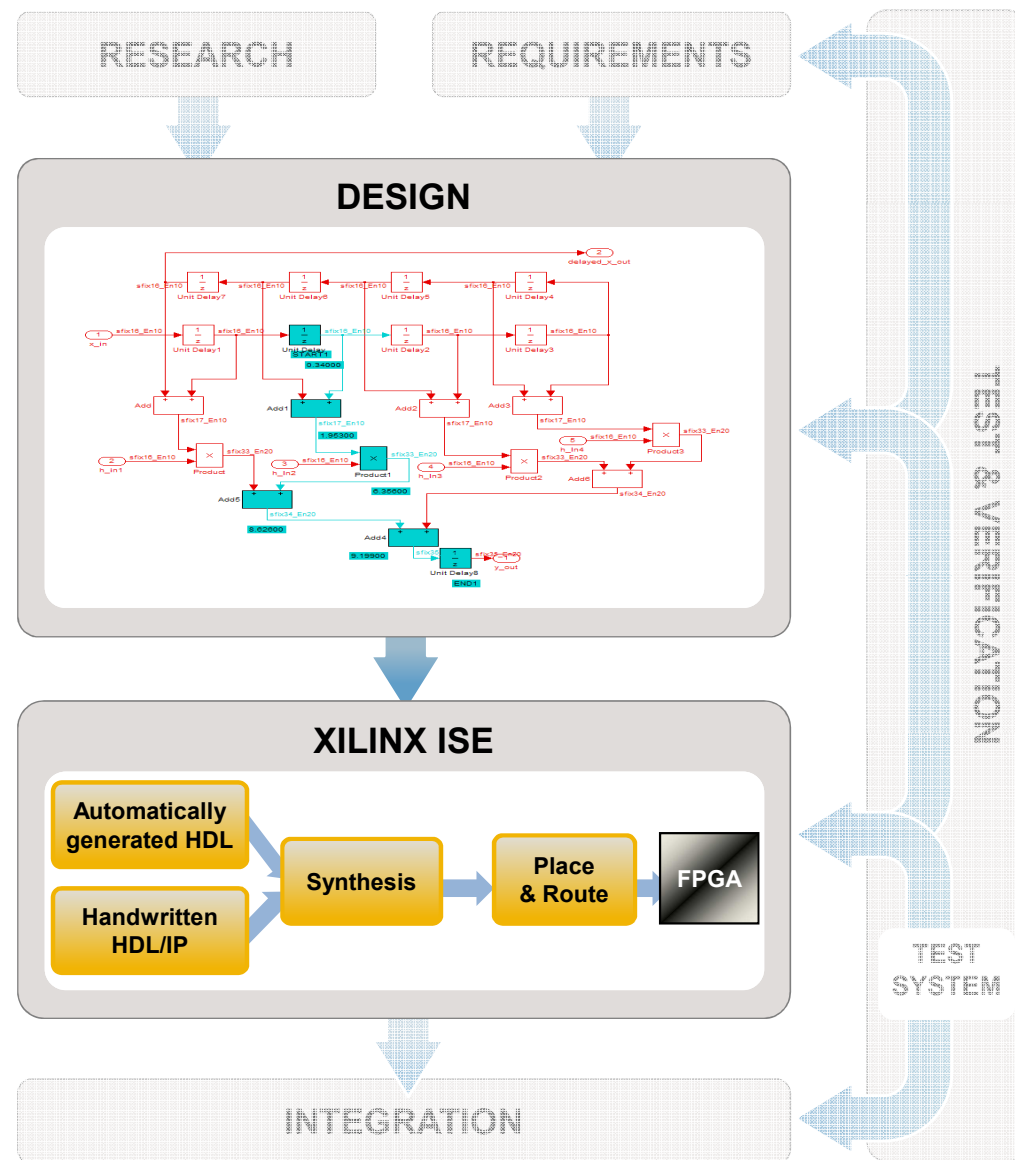
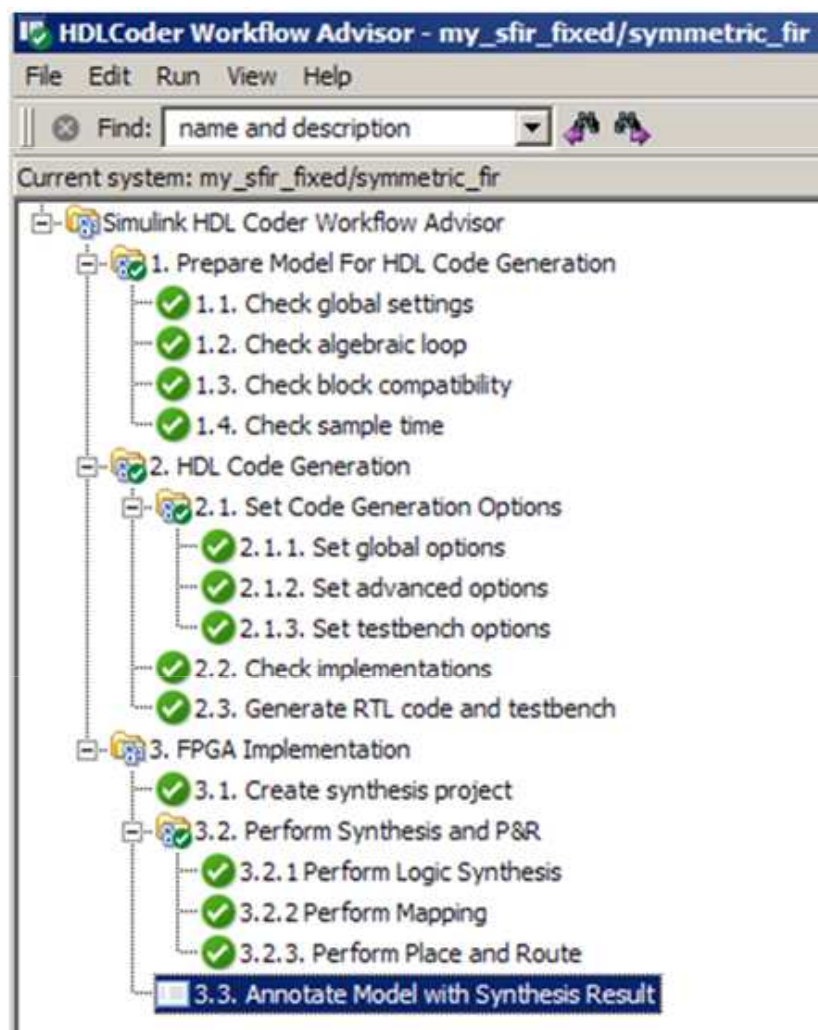


Как быстро перенести идеи в «железо»?

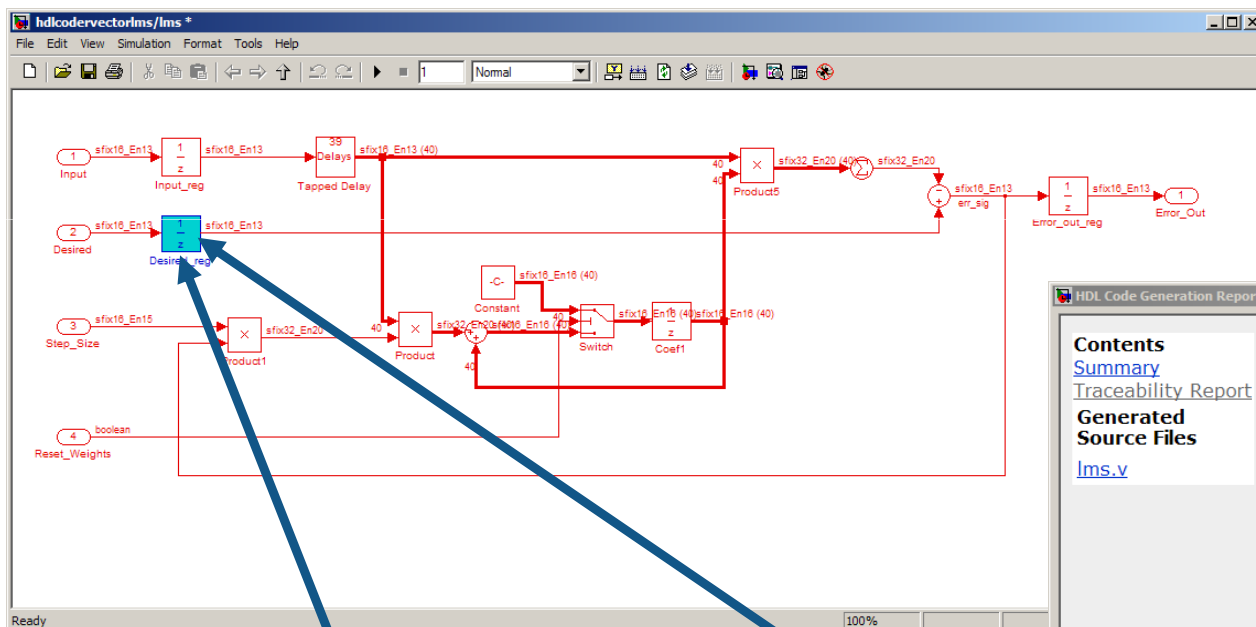
Решения от MathWorks



Разработка на ПЛИС компании XILINX



Проверка на соответствие стандарту DO-254



```
621 assign Constant_out1[38] = 0;
622 assign Constant_out1[39] = 0;
623
624
625
626 // <S11>/Desired_reg
627 always @(posedge clk or posedge reset)
628     begin : Desired_reg_process
629         if (reset)
630             Desired_reg_out1 <= 0;
631         else
632             if (enb)
633                 Desired_reg_out1 <= Desired;
634
635     end
```

Traceability Report for hdlcodervectorlms

Table of Contents

1. [Eliminated / Virtual Blocks](#)
2. [Traceable Simulink Blocks / Stateflow Objects / Embedded MATLAB Scripts](#)
 - o [hdlcodervectorlms/lms](#)

Eliminated / Virtual Blocks

Block Name	Comment
<S11>/Input	Import
<S11>/Desired	Import
<S11>/Step_Size	Import
<S11>/Reset_Weights	Import
<S11>/Error_Out	Outport

Traceable Simulink Blocks / Stateflow Objects / Embedded MATLAB Scripts

Subsystem: [hdlcodervectorlms/lms](#)

Object Name	Code Location
<S11>/Coef1	lms.v:1137
<S11>/Constant	lms.v:582
<S11>/Desired_reg	lms.v:626
<S11>/Error_out_reg	lms.v:1608
<S11>/Input_reg	lms.v:438
<S11>/Product	lms.v:645

Темы доклада

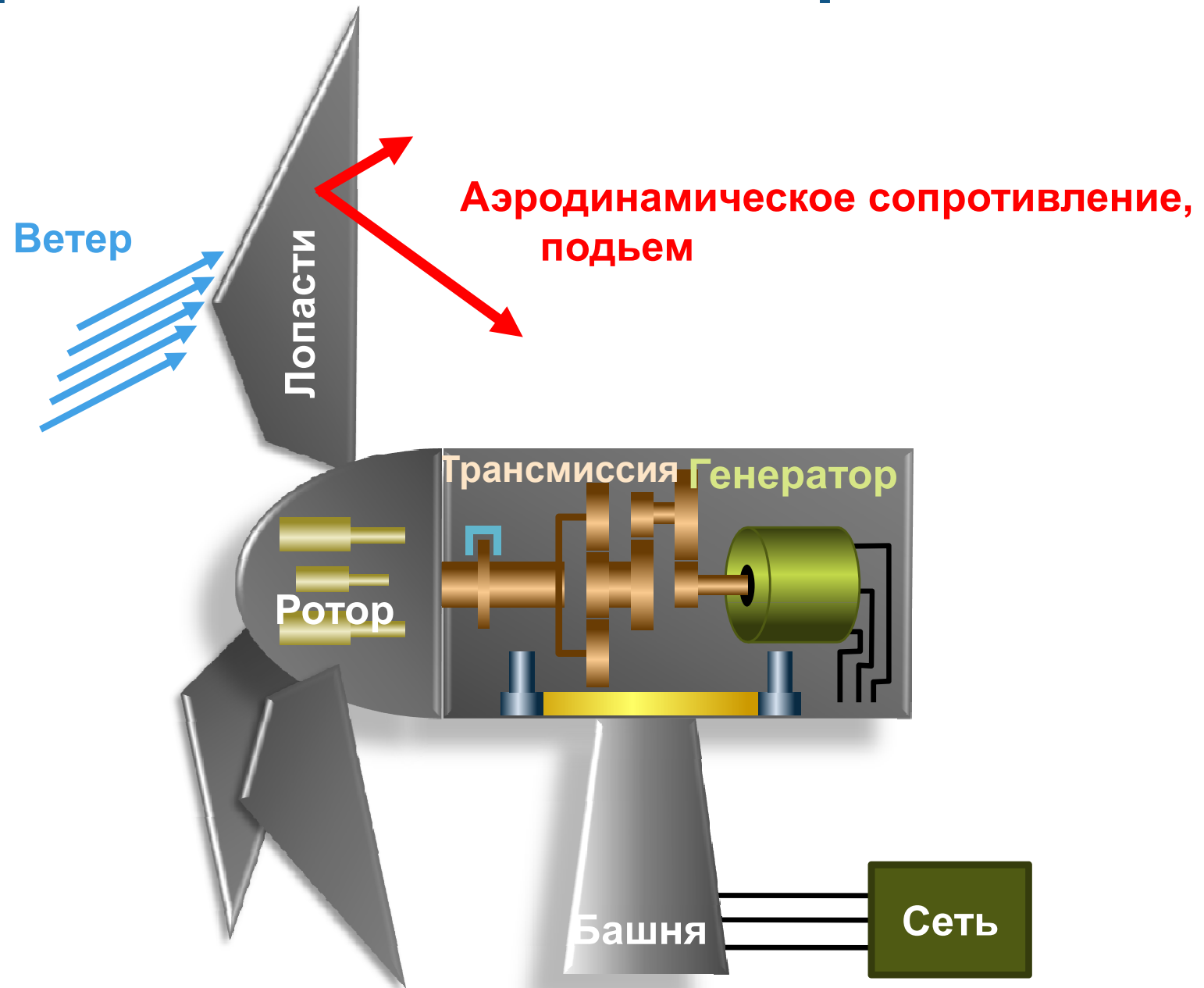
- ПЛИС и встраиваемые системы
 - Simulink HDL Coder
- Совместное моделирование различных физических доменов
- Упрощенная разработка систем управления
- Генерация C кода из Matlab



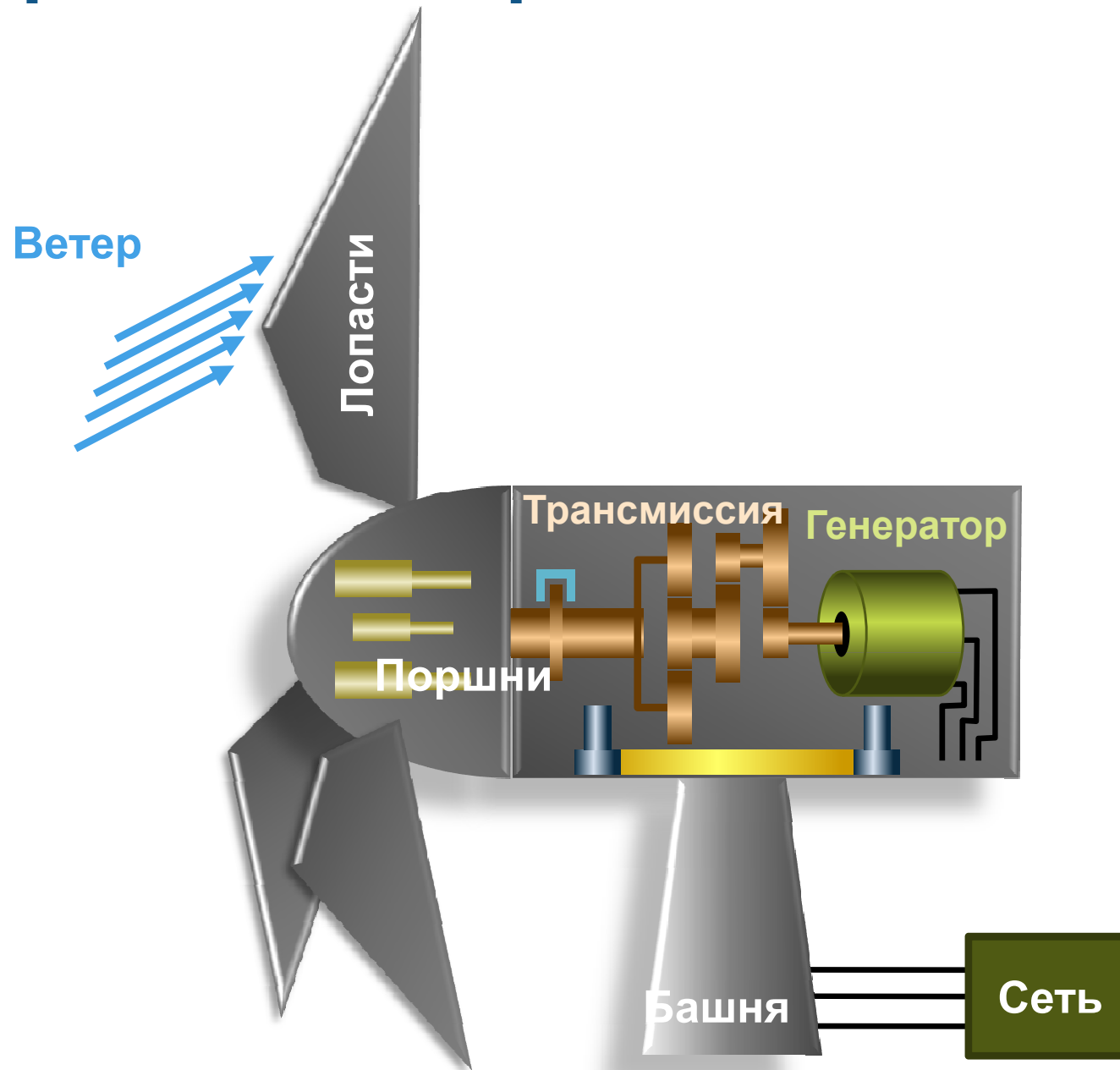
Совместное моделирование различных физических доменов



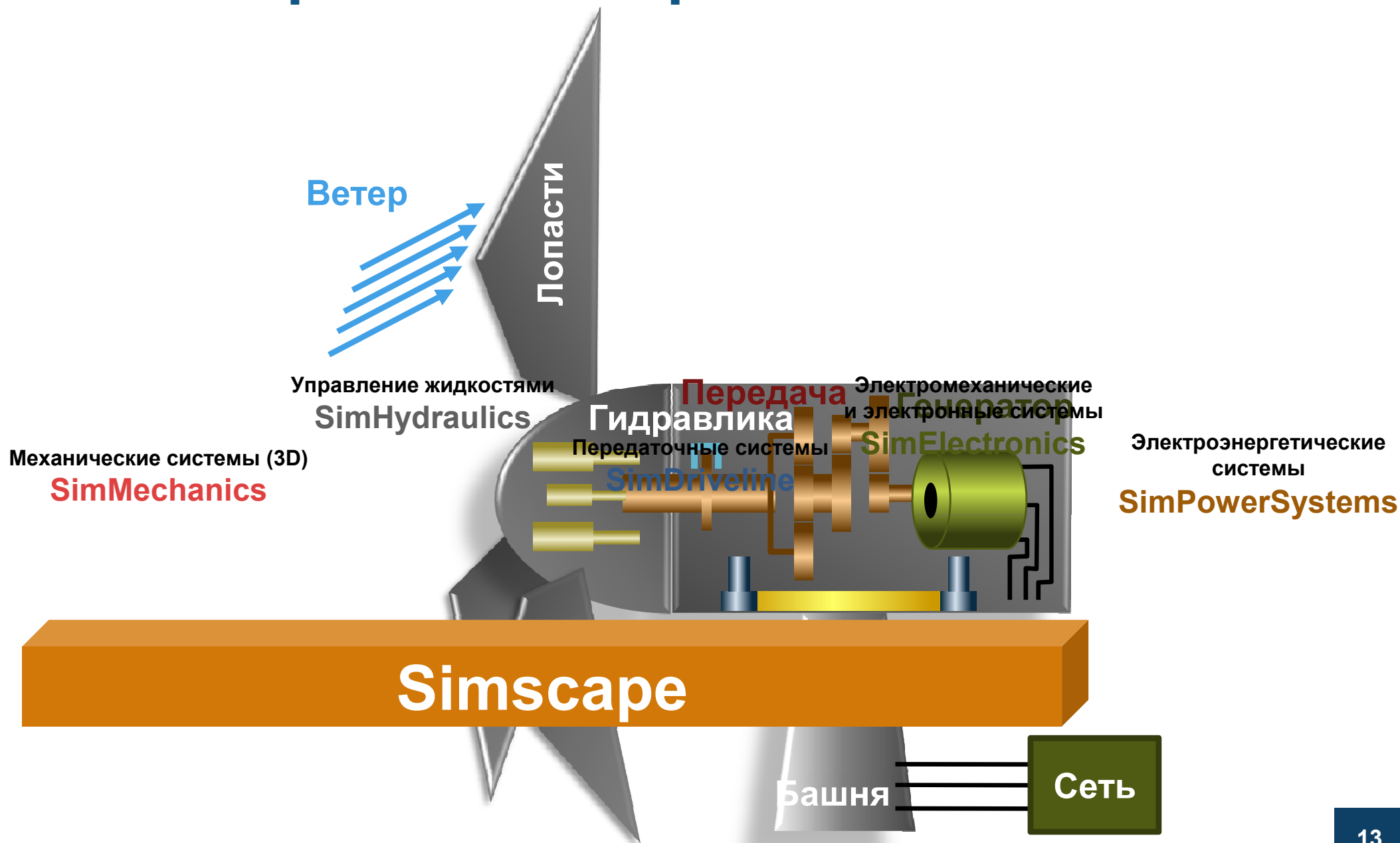
Пример совместного моделирования



Охват различных физических систем



Охват различных физических систем

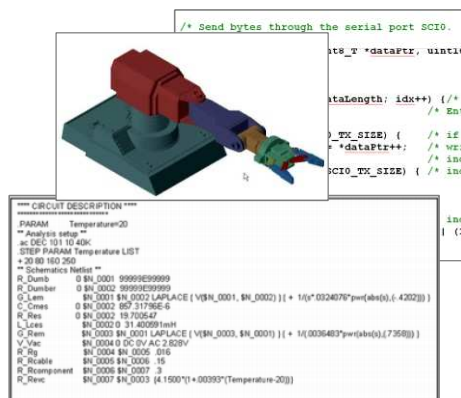
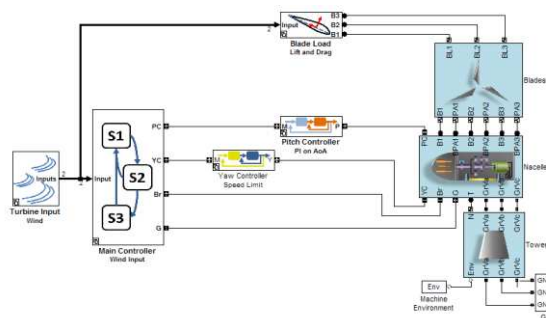


Решения MathWorks

Исследования

Требования

Разработка



Требования соединены с моделью

Модель детализируется описывая различные физические процессы

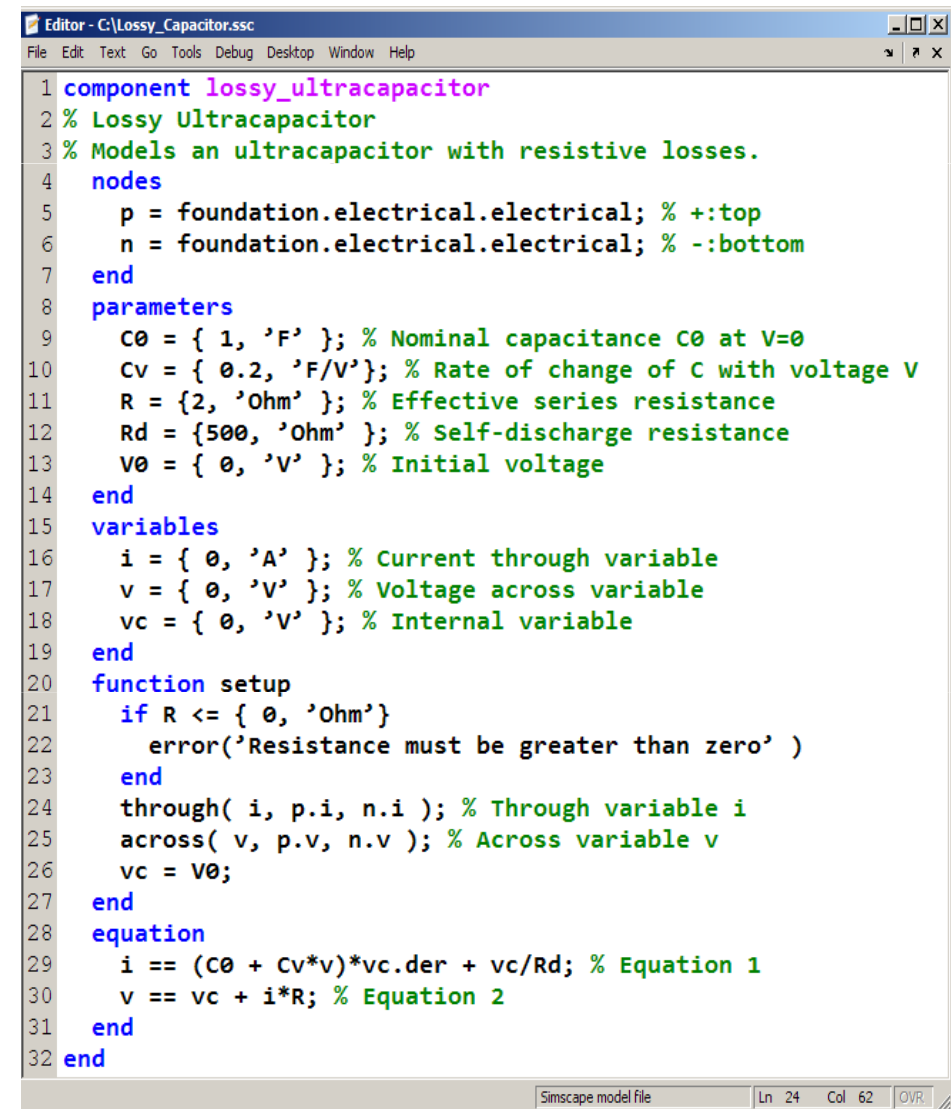
Существующие проекты совместно используются с моделью (САПР, метод конечных элементов и SPICE модели)

Моделирование различных физических процессов в единой среде разработки

Язык Simscape

Это язык основанный на MATLAB, позволяющий описывать физические компоненты, физические домены и библиотеки.

- Использует все возможности MATLAB
- Позволяет применять объектно-ориентированный подход
- Генерация Simulink блоков
- Возможен перевод в двоичный файл, для защиты интеллектуальной собственности



```
Editor - C:\Lossy_Capacitor.ssc
File Edit Text Go Tools Debug Desktop Window Help

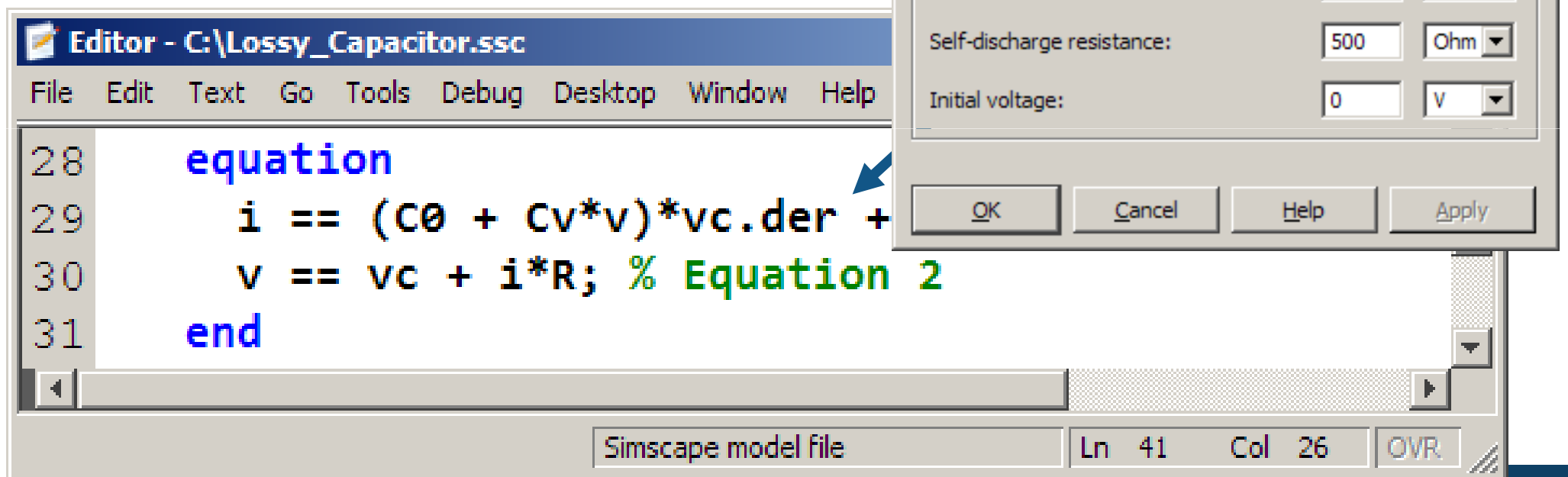
1 component lossy_ultracapacitor
2 % Lossy Ultracapacitor
3 % Models an ultracapacitor with resistive losses.
4 nodes
5     p = foundation.electrical.electrical; % +:top
6     n = foundation.electrical.electrical; % -:bottom
7 end
8 parameters
9     C0 = { 1, 'F' }; % Nominal capacitance C0 at V=0
10    Cv = { 0.2, 'F/V' }; % Rate of change of C with voltage V
11    R = { 2, 'Ohm' }; % Effective series resistance
12    Rd = { 500, 'Ohm' }; % Self-discharge resistance
13    V0 = { 0, 'V' }; % Initial voltage
14 end
15 variables
16     i = { 0, 'A' }; % Current through variable
17     v = { 0, 'V' }; % Voltage across variable
18     vc = { 0, 'V' }; % Internal variable
19 end
20 function setup
21     if R <= { 0, 'Ohm' }
22         error('Resistance must be greater than zero' )
23     end
24     through( i, p.i, n.i ); % Through variable i
25     across( v, p.v, n.v ); % Across variable v
26     vc = V0;
27 end
28 equation
29     i == (C0 + Cv*v)*vc.der + vc/Rd; % Equation 1
30     v == vc + i*R; % Equation 2
31 end
32 end

Simscape model file Ln 24 Col 62 OVR
```

Пример использования Simscape

Уравнения определяются с
помощью текстового
описания

$i =$



The screenshot displays the Simscape environment. On the right, the 'Block Parameters: Lossy Ultracapacitor' dialog box is open, showing the following parameters:

- Nominal capacitance C0 at V=0: 1 F
- Rate of change of C with voltage V: 0.2 F/V
- Effective series resistance: 2 Ohm
- Self-discharge resistance: 500 Ohm
- Initial voltage: 0 V

Below the parameters are buttons for OK, Cancel, Help, and Apply. To the left, the 'Editor - C:\Lossy_Capacitor.ssc' window shows the following code:

```
28 equation
29     i == (C0 + Cv*v)*vc.der +
30     v == vc + i*R; % Equation 2
31 end
```

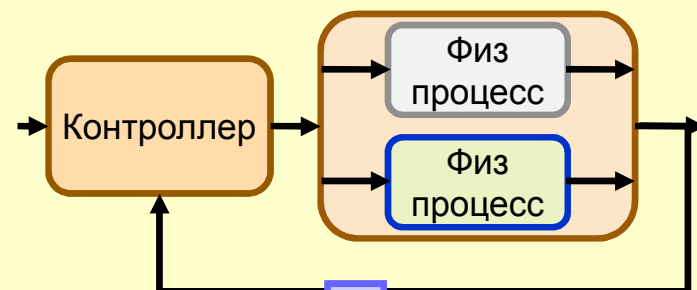
A blue arrow points from the 'i ==' line in the code to the 'i =' text above the dialog box. The status bar at the bottom indicates 'Simscape model file', 'Ln 41', 'Col 26', and 'OVR'.

Выбор решателя улучшает производительность

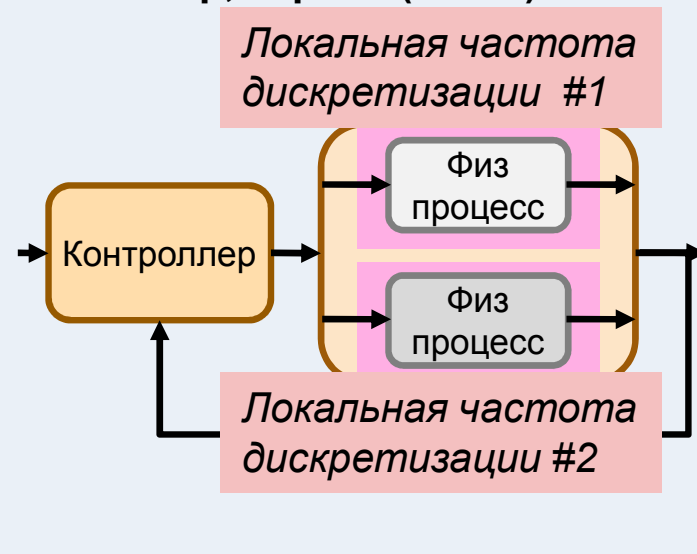
Используйте локальный решатель с фиксированным шагом для моделирования физических процессов.

- Запуск моделирования физических процессов с разной частотой дискретизации
- Улучшается производительность

Variable step, implicit (ODE15s)



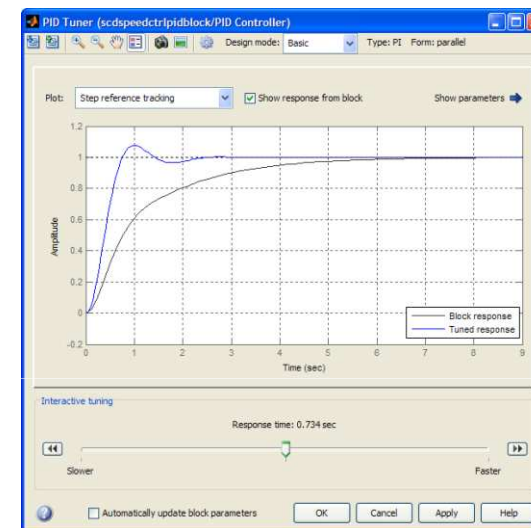
Fixed step, explicit (ODE1)



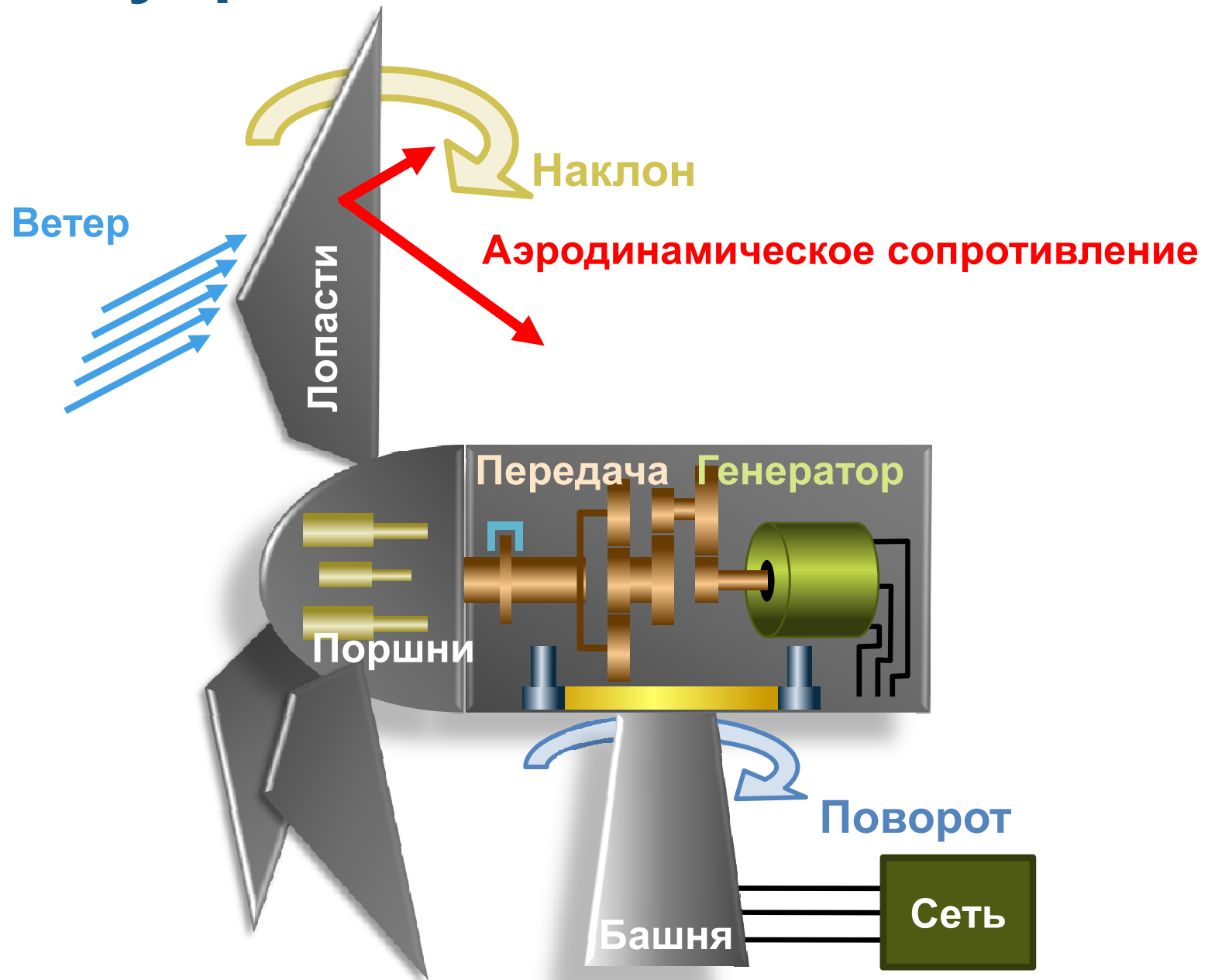
Генерация кода

Темы доклада

- ПЛИС и встраиваемые системы
 - Simulink HDL Coder
- Совместное моделирование различных физических доменов
 - язык Simscape
- Упрощенная разработка систем управления
- Генерация С кода из MATLAB



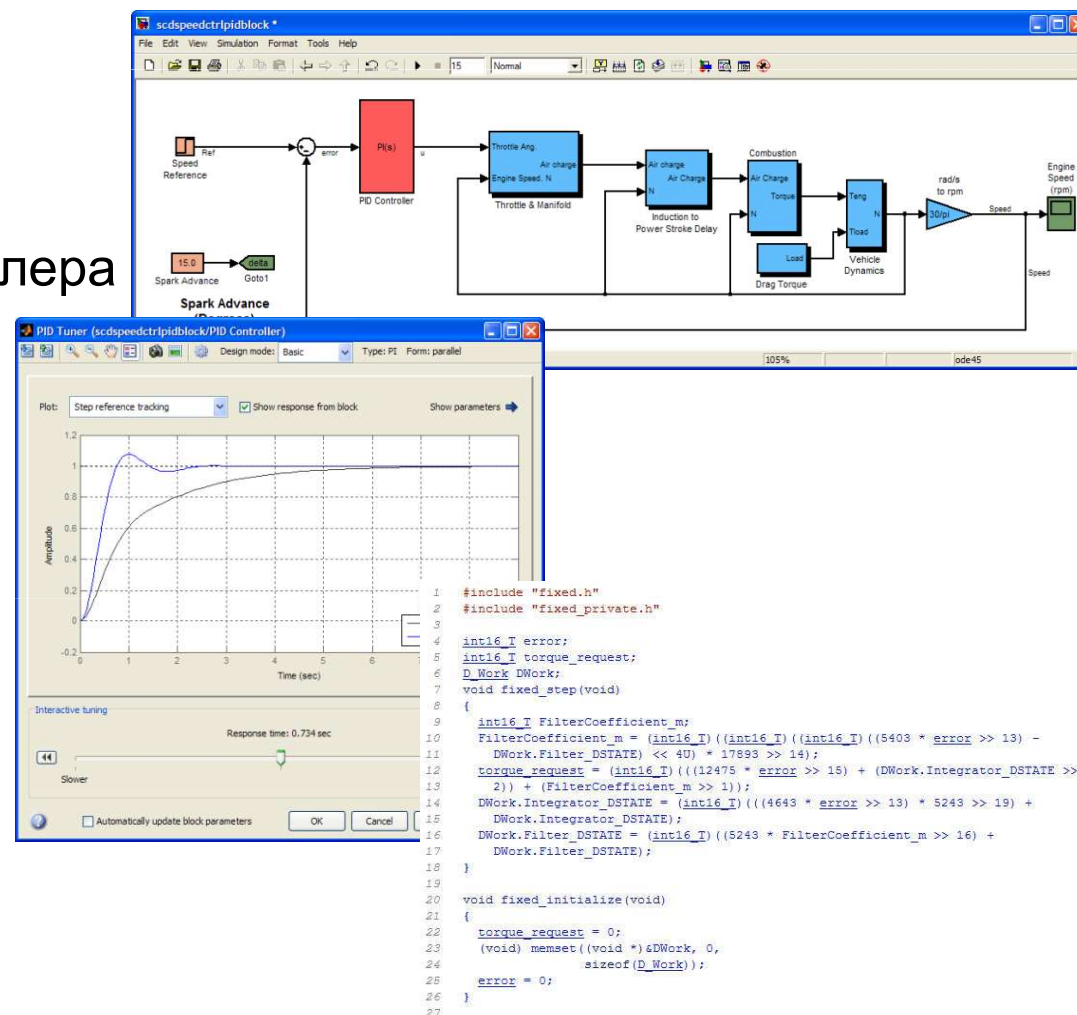
Системы управления



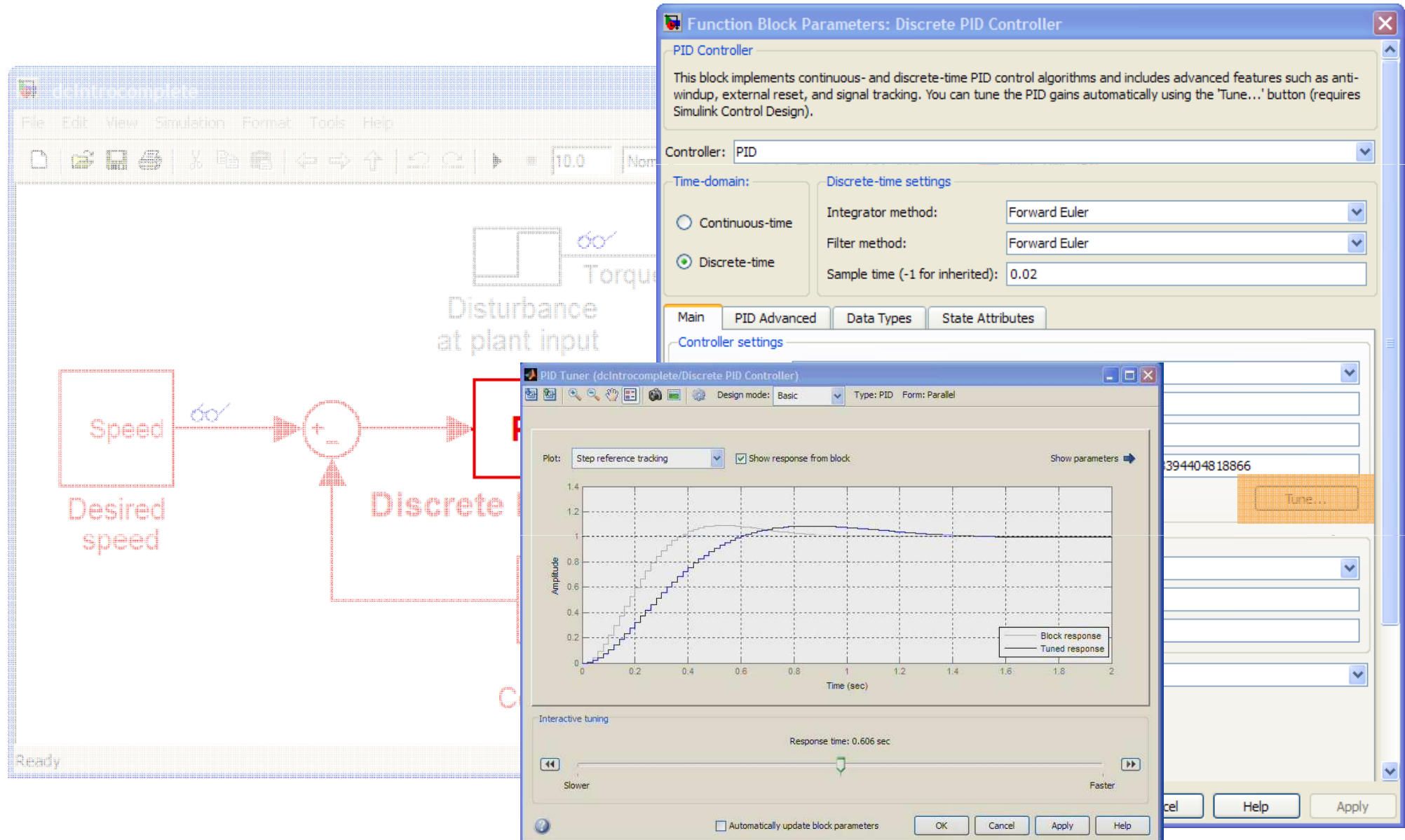
Разработка ПИД контроллеров стала проще

Новый блок ПИД контроллера промышленного уровня с автоматической настройкой

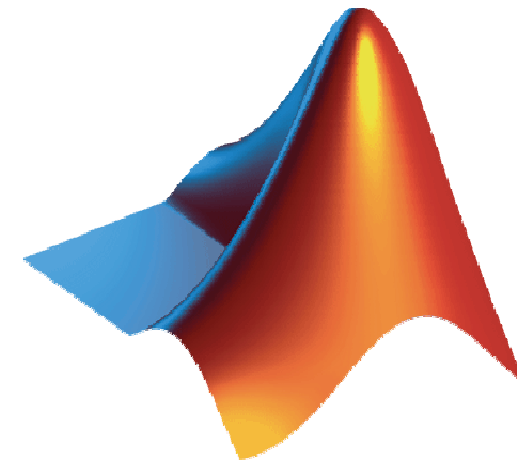
- Различные формы ПИД контроллера
- Интегратор с фиксатором выходного сигнала
- Автоматическая настройка для достижения соответствия требованиям
- Дополнительная интерактивная подстройка
- Перевод в арифметику с фиксированной точкой
- Генерация C и PLC кода



Автоматическая подстройка

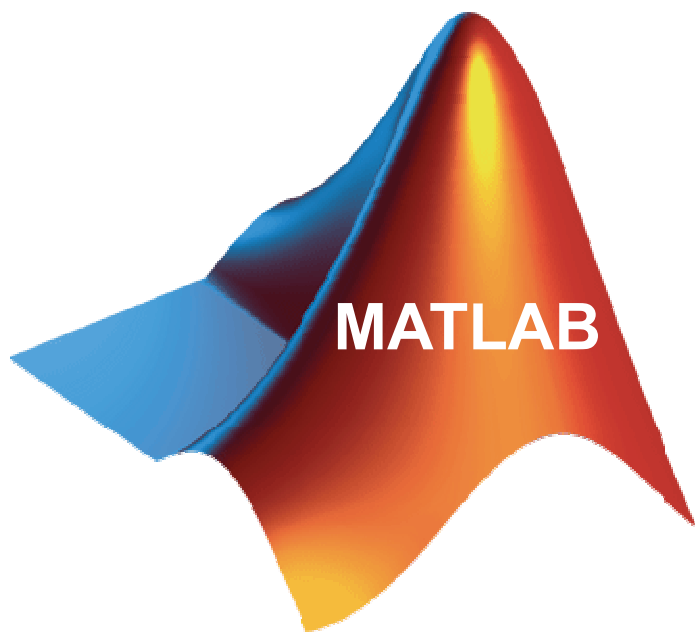


Темы доклада



- ПЛИС и встраиваемые системы
 - Simulink HDL Coder
- Совместное моделирование различных физических процессов
 - язык Simscape
- Как облегчить разработку систем управления
 - ПИД контроллер с автонастройкой
- Генерация C кода из MATLAB

Генерация С кода из MATLAB

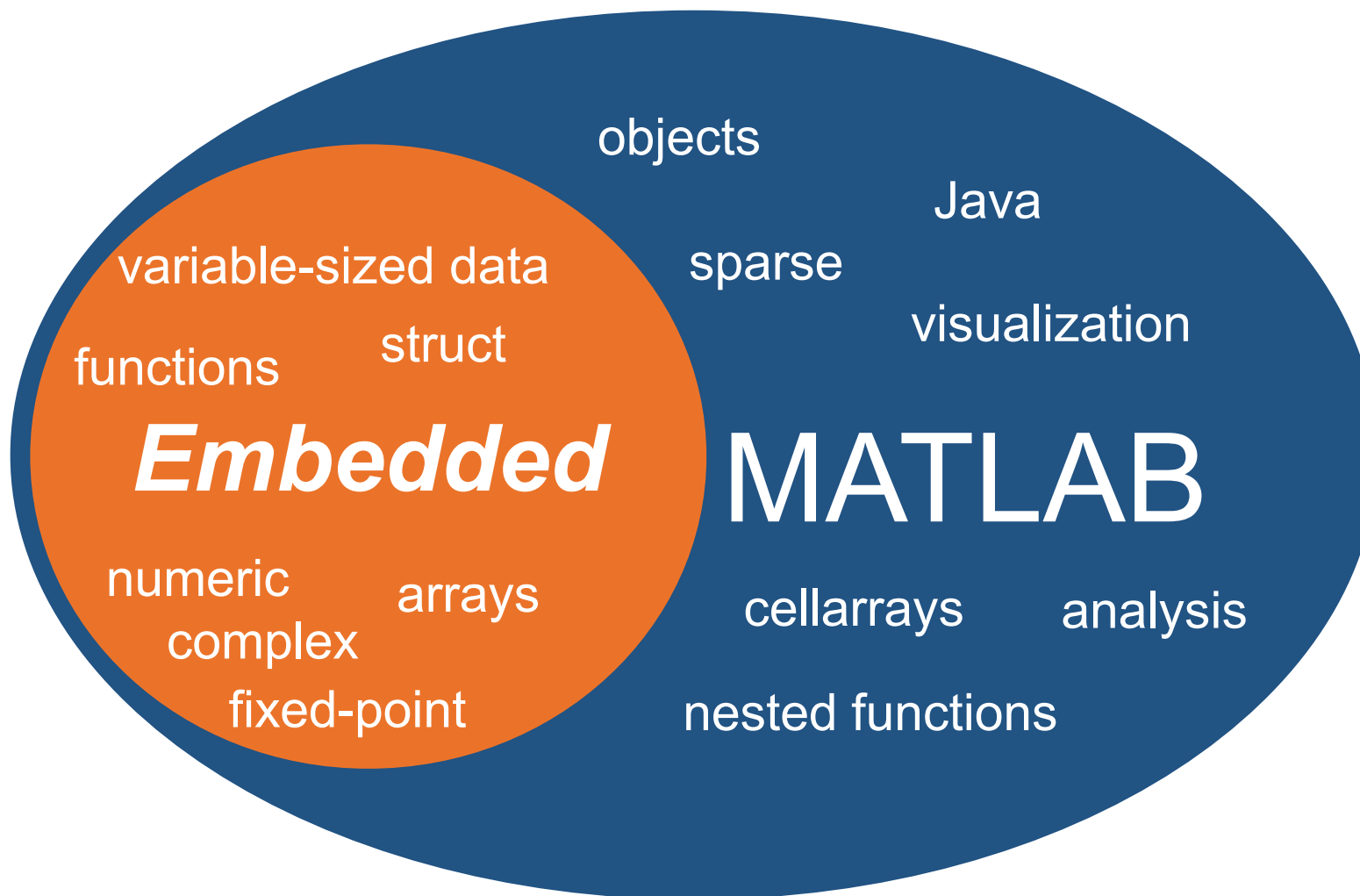


→ C/C++

Автоматический перевод MATLAB
кода в C/C++ код

- Быстрое прототипирование
- Проверка алгоритма на адекватность в реальных условиях

Решение MathWorks

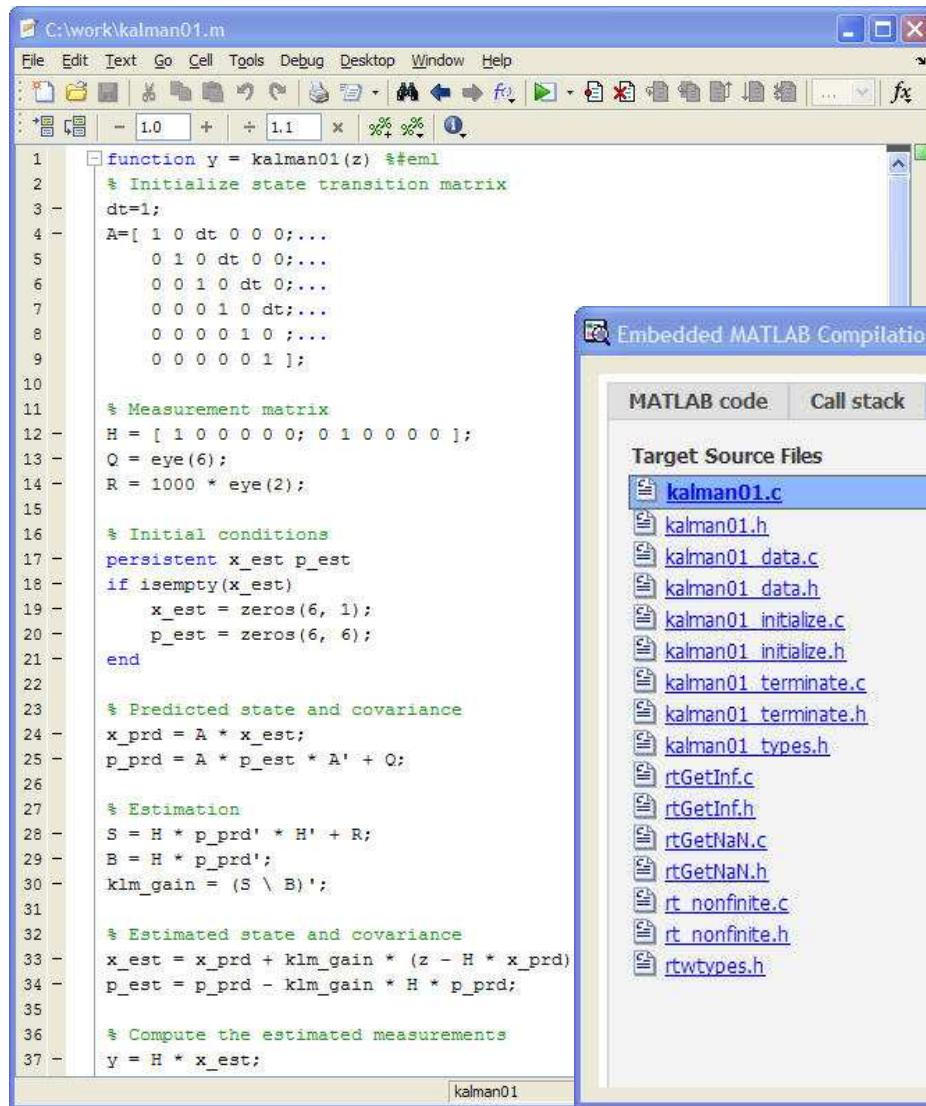


Embedded MATLAB – это подмножество MATLAB для разработки встраиваемых систем

Embedded MATLAB

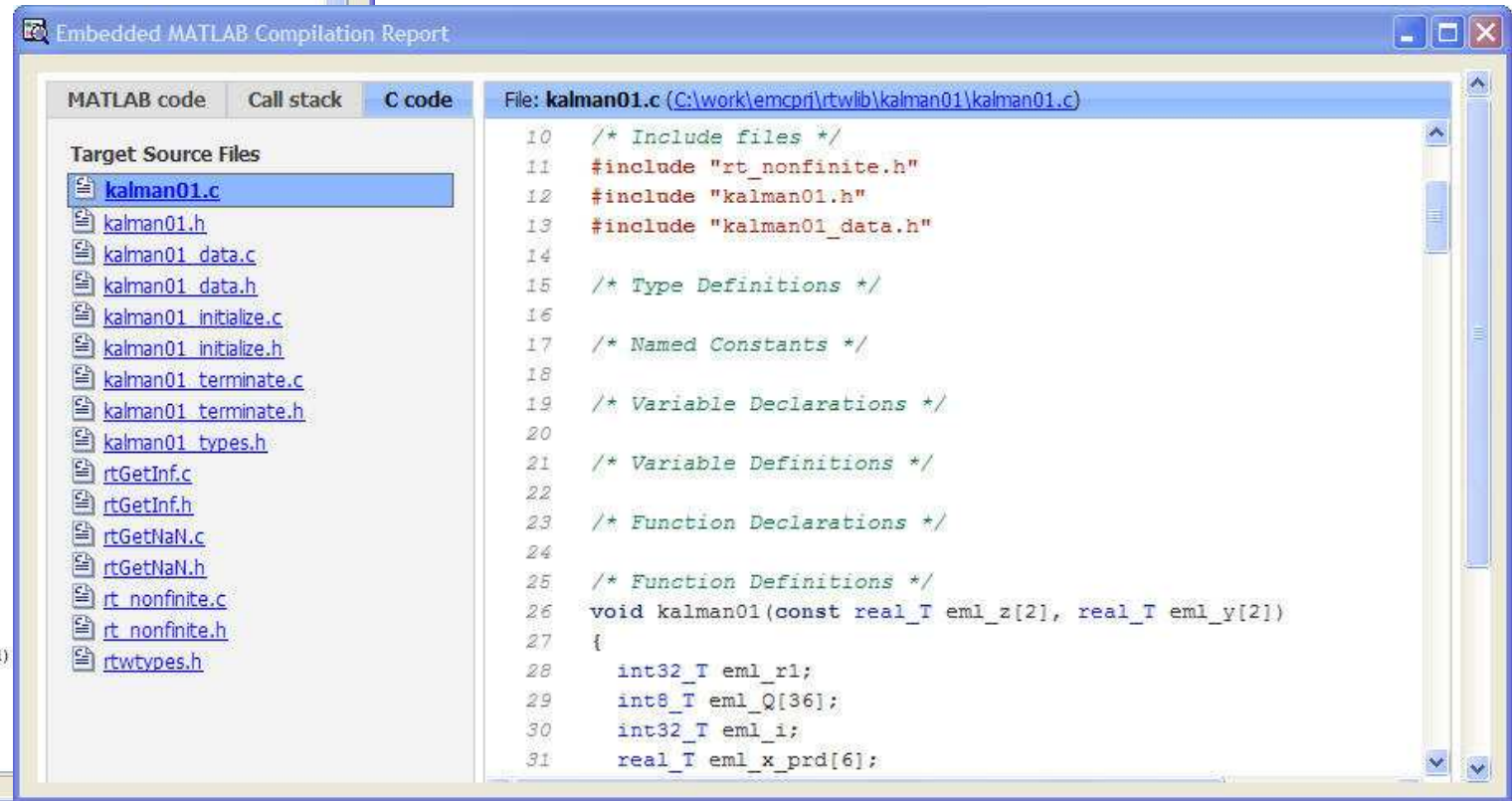
- **Один язык**
 - Нет нескольких копий исходных кодов
 - Нет повторяющихся операций тестирования и отладки кода
- **Одна среда разработки**
 - Интеграция с моделями окружения описанными в MATLAB
 - Интеграция с визуализацией, анализом и отладкой
 - Простое портирование алгоритмов MATLAB
- **Автоматическая генерация кода**
 - Возможность получения встроенного кода (MATLAB to C)

Пример Embedded MATLAB (MATLAB)



```
1 function y = kalman01(z) %#eml
2 % Initialize state transition matrix
3 dt=1;
4 A=[ 1 0 dt 0 0 0;...
5     0 1 0 dt 0 0;...
6     0 0 1 0 dt 0;...
7     0 0 0 1 0 dt;...
8     0 0 0 0 1 0 ;...
9     0 0 0 0 0 1 ];
10
11 % Measurement matrix
12 H = [ 1 0 0 0 0 0; 0 1 0 0 0 0 ];
13 Q = eye(6);
14 R = 1000 * eye(2);
15
16 % Initial conditions
17 persistent x_est p_est
18 if isempty(x_est)
19     x_est = zeros(6, 1);
20     p_est = zeros(6, 6);
21 end
22
23 % Predicted state and covariance
24 x_prd = A * x_est;
25 p_prd = A * p_est * A' + Q;
26
27 % Estimation
28 S = H * p_prd' * H' + R;
29 B = H * p_prd';
30 klm_gain = (S \ B)';
31
32 % Estimated state and covariance
33 x_est = x_prd + klm_gain * (z - H * x_prd);
34 p_est = p_prd - klm_gain * H * p_prd;
35
36 % Compute the estimated measurements
37 y = H * x_est;
```

>> emlc -eg {z} kalman.m



Embedded MATLAB Compilation Report

MATLAB code | Call stack | C code

File: kalman01.c (C:\work\emcpri\rtwlib\kalman01\kalman01.c)

Target Source Files

- kalman01.c
- kalman01.h
- kalman01_data.c
- kalman01_data.h
- kalman01_initialize.c
- kalman01_initialize.h
- kalman01_terminate.c
- kalman01_terminate.h
- kalman01_types.h
- rtGetInf.c
- rtGetInf.h
- rtGetNaN.c
- rtGetNaN.h
- rt_nonfinite.c
- rt_nonfinite.h
- rtwtypes.h

```
10 /* Include files */
11 #include "rt_nonfinite.h"
12 #include "kalman01.h"
13 #include "kalman01_data.h"
14
15 /* Type Definitions */
16
17 /* Named Constants */
18
19 /* Variable Declarations */
20
21 /* Variable Definitions */
22
23 /* Function Declarations */
24
25 /* Function Definitions */
26 void kalman01(const real_T eml_z[2], real_T eml_y[2])
27 {
28     int32_T eml_r1;
29     int8_T eml_Q[36];
30     int32_T eml_i;
31     real_T eml_x_prd[6];
```

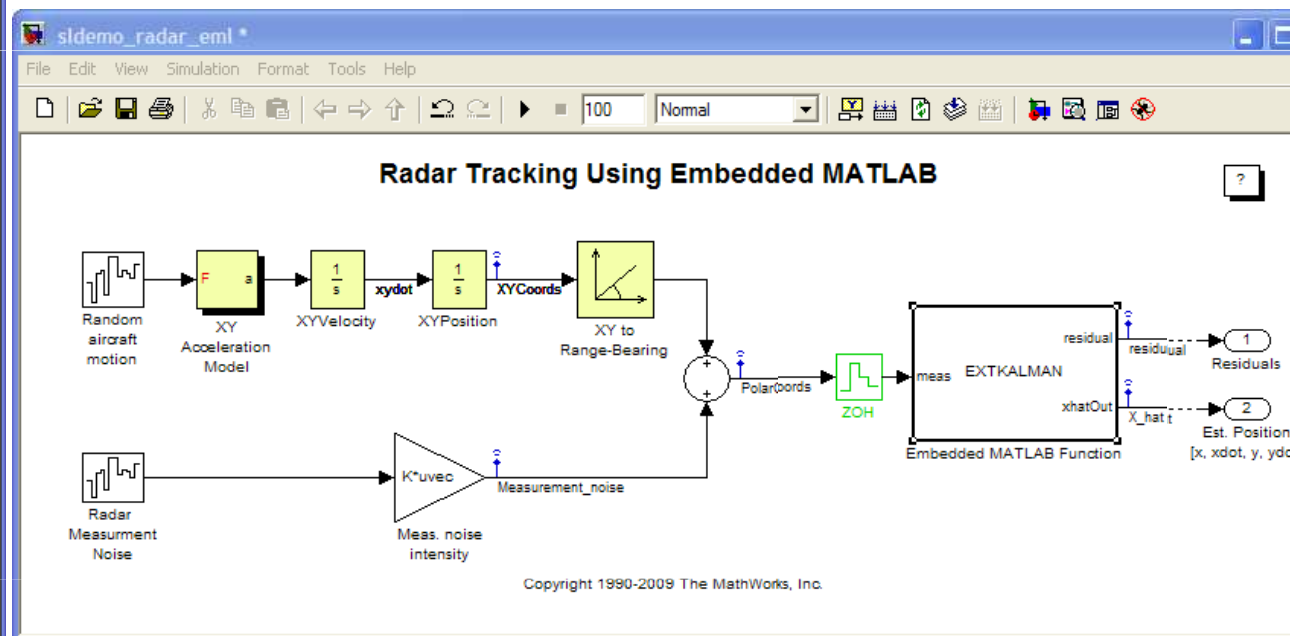
Перевод кода MATLAB в код C/C++

Пример Embedded MATLAB (Simulink)

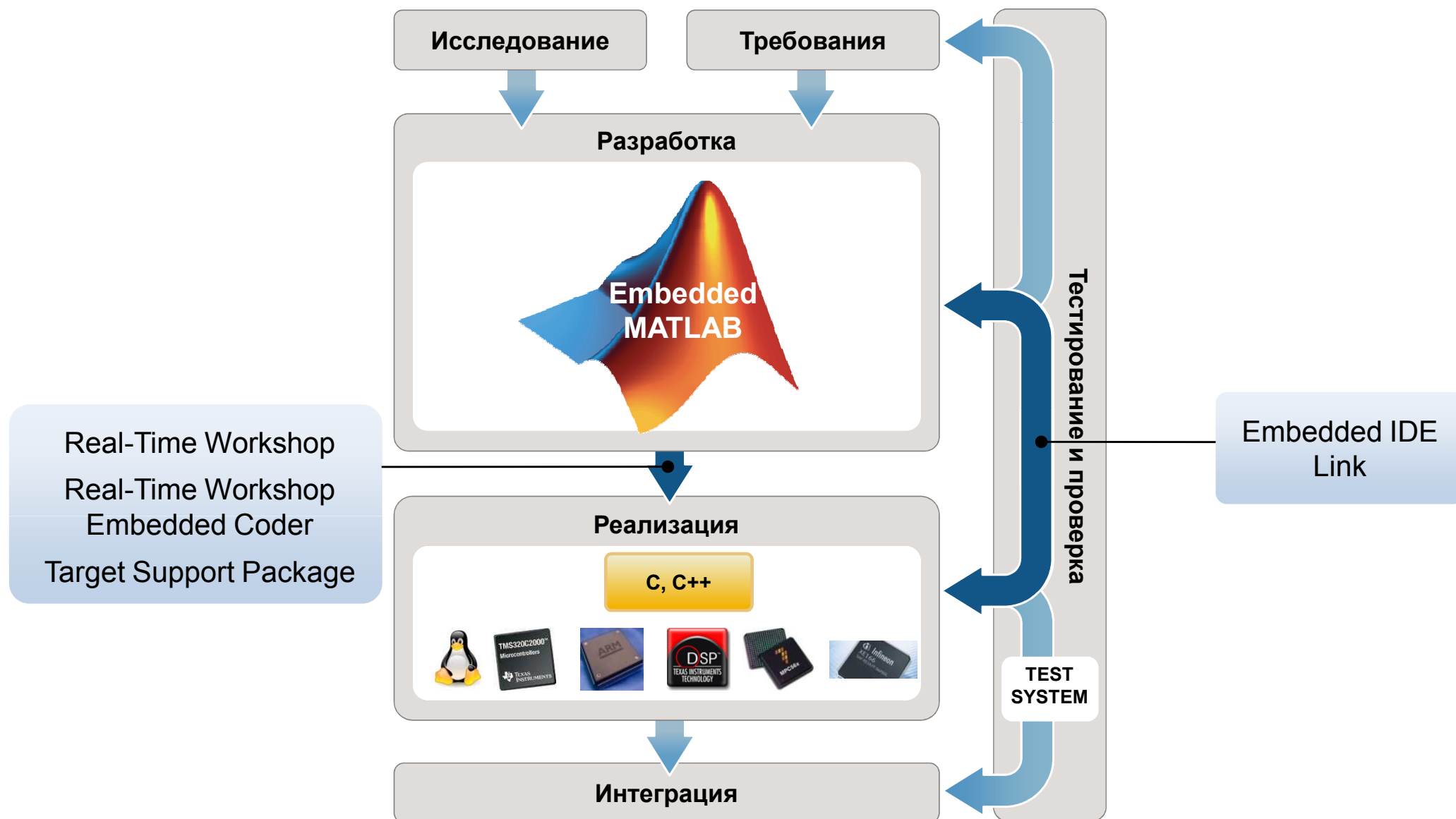
```

Embedded MATLAB Editor - Block: sldemo_radar_eml/Embedded MATLAB Function
File Edit Text Debug Tools Window Help
20 % Initialization
21 persistent P;
22 persistent xhat
23 if isempty(P)
24     xhat = [0.001; 0.01; 0.001; 400];
25     P = zeros(4);
26 end
27
28 % Radar update time deltat is inherited from model workspace
29
30 % 1. Compute Phi, Q, and R
31 Phi = [1 deltat 0 0; 0 1 0 0; 0 0 1 deltat; 0 0 0 1];
32 Q = diag([0 .005 0 .005]);
33 R = diag([300^2 0.001^2]);
34
35 % 2. Propagate the covariance matrix:
36 P = Phi*P*Phi' + Q;
37
38 % 3. Propagate the track estimate::
39 xhat = Phi*xhat;
40
41 % 4 a). Compute observation estimates:
42 Rangehat = sqrt(xhat(1)^2+xhat(3)^2);
43 Bearinghat = atan2(xhat(3),xhat(1));
44
45 % 4 b). Compute observation vector y and linearized measurement matrix M
46 yhat = [Rangehat;
47         Bearinghat];
48 M = [ cos(Bearinghat)      0 sin(Bearinghat)      0
49       -sin(Bearinghat)/Rangehat 0 cos(Bearinghat)/Rangehat 0 ];
50
51 % 4 c). Compute residual (Estimation Error)
52 residual = meas - yhat;
53
54 % 5. Compute Kalman Gain:
55 W = P*M'*inv(M*P*M' + R);
56
57 % 6. Update estimate
58 xhat = xhat + W*residual;
59
60 % 7. Update Covariance Matrix
61 P = (eye(4) - W*M)*P*(eye(4) - W*M)' + W*R*W';
62
63 xhatOut = xhat;
64
Ready Ln 1 Col 1

```



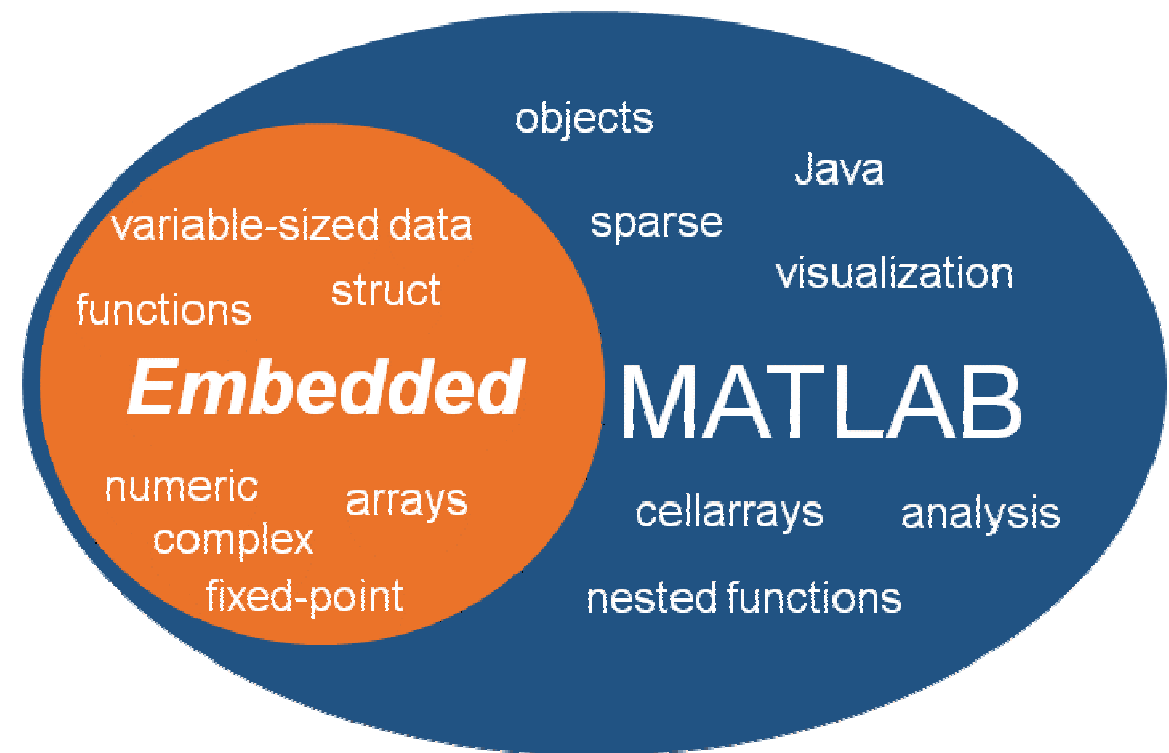
Решения для встроенного кода



Расширение функциональности Embedded MATLAB

Расширение
функциональности под
типичные алгоритмы для
встраиваемых систем

- Типы данных с фиксированной точкой
- Алгоритмы
 - Обработка сигналов
 - Обработка видео
- Параметры с различным размером



Резюме

- ПЛИС и встраиваемые системы
 - Simulink HDL Coder
- Совместное моделирование различных физических процессов
 - язык Simscape
- Упрощенная разработка систем управления
 - ПИД контроллер с автонастройкой
- Генерация С кода из MATLAB
 - Embedded MATLAB

