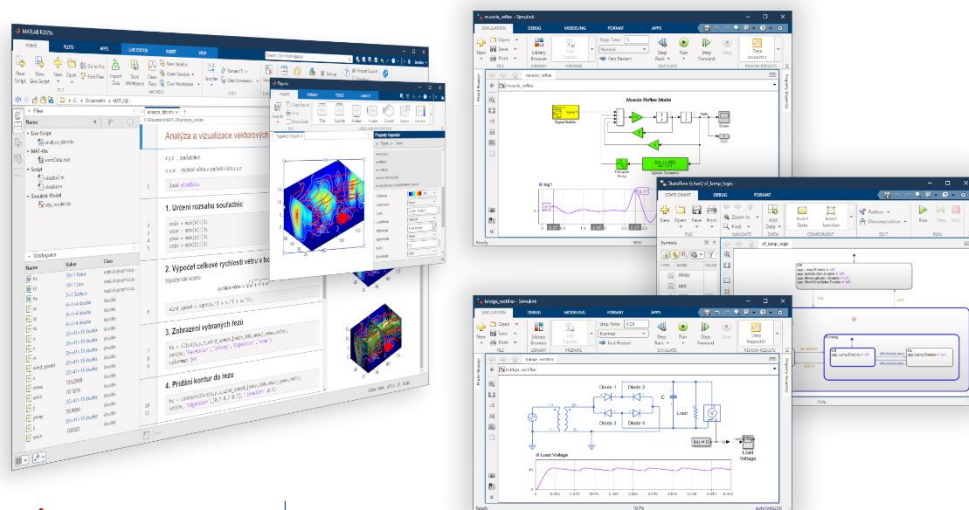


10.9.2026 Technical Computing Camp 2026

Nové nástroje, materiály a služby v oblasti vzdělávání



Michal Blaho

blaho@humusoft.sk

Martin Foltin

foltin@humusoft.sk

www.humusoft.cz

info@humusoft.cz

www.mathworks.com

Campus-Wide License



Campus-Wide License

Accelerate learning and research with MATLAB and Simulink

[Request a quote](#)

A [Campus-Wide License](#) provides unlimited use of MATLAB and Simulink products to all faculty, staff, students, and researchers, on and off campus, on any device.

Supported Institutions and Programs

- Research universities
- Teaching universities and colleges
- Business schools
- Medical schools
- Liberal arts colleges

More than 2,400 schools already have a Campus-Wide License. To gain access to MATLAB and Simulink, create a MathWorks Account using your university email address.

[Get started](#)

MATLAB Campus-Wide licence v ČR a SR



MATLAB CourseWare

- Interaktívny obsah a príklady
 - vytvorený MathWorks a učiteľmi
- Obsah podľa typu
 - aplikácie, príklady, laboratória
- Obsah podľa disciplín
 - riadenie, štatistika, biológia
- Úroveň vzdelania
 - základné a stredné školy
 - univerzity
- Ďalšie zdroje na GitHubu
 - [MathWorks Teaching Resources](#)

Control Systems - University of Cape Town

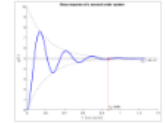
An introductory control systems course, which is intended to delivery fundamental concepts with the use of interactive MATLAB Live Scripts.

Downloads (30 days)



Courseware Type: Interactive Examples, Labs, Projects

Education Levels: Undergraduate



Machine Learning for Regression

Interactive module that introduces typical workflow, setup, and considerations involved in solving regression problems with machine learning

Downloads (30 days)



Courseware Type: Interactive Examples

Education Levels: Undergraduate



Applied Partial Differential Equations

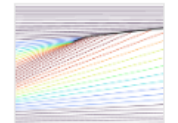
A collection of teaching scripts applying partial differential equations to traffic modeling, battery thermal modeling, diffusion, and more.

Downloads (30 days)



Courseware Type: Interactive Examples

Education Levels: Graduate, Undergraduate



Calculus Derivatives

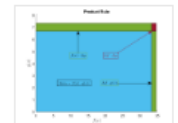
Interactive module that introduces fundamentals of derivatives including the product and chain rule as presented in Calculus I courses

Downloads (30 days)



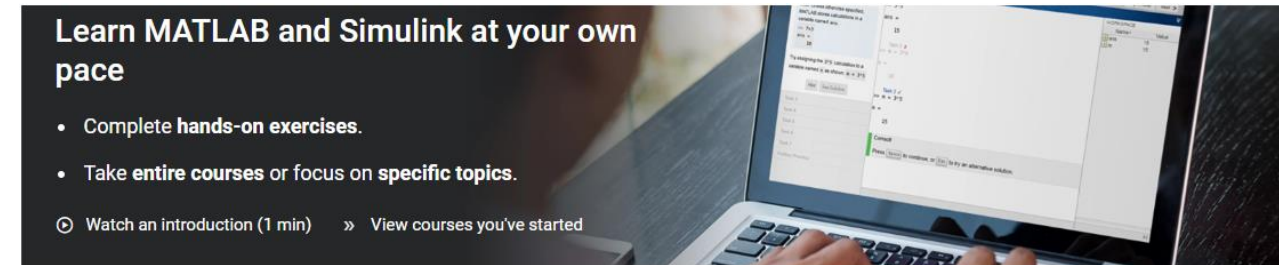
Courseware Type: Interactive Examples

Education Levels: Undergraduate



Sada kurzov Online Training Suite

- Sada online kurzov
 - MATLAB Academy
- K dispozícii kedykoľvek a kdekoľvek
 - stačí počítač s webovým prehliáčom
- Vlastným tempom
 - možnosť kurz kedykoľvek prerušiť
- Priebežné kvízy
 - okamžitá spätná väzba k riešeniu úloh
- Certifikát o absolvovaní kurzu
 - pdf alebo online odkaz
- Pre učiteľov možnosť „sledovať pokrok“ študentov v kurze



FILTER BY

Type

Course	68
Onramp (Introductory Course)	24
Learning Path	12

Using MATLAB

All MATLAB	45
Get Started with MATLAB	6
Language Fundamentals	15
Data Import and Analysis	17
Graphics	5
Programming	14
App Building	1
Software Development Tools	8

Sort by Featured

Results 1 - 10 of 80



New to MATLAB? Start here:
MATLAB Onramp
 2 hours | [Languages](#)
 Get started quickly with the basics of MATLAB.



New to Simulink? Start here:
Simulink Onramp
 2 hours | [Languages](#)
 Get started quickly with the basics of Simulink.



Core MATLAB Skills
LEARNING PATH: 4 COURSES
 Develop your core MATLAB skills to tackle new challenges with confidence.

Build Simulink Proficiency

- Kurz o efektívnom využívaní prostredia Simulink

- krátke videá a praktické cvičenia
- automatizované hodnotenia
- okamžitá spätná väzba

- Témy zahŕňajú:

- základy Simulinku
- modelovanie dynamiky a organizácia modelu
- opakovane použiteľné komponenty
- zlepšenie výkonu simulácie



The Simulink Environment



Model Dynamic Systems



Create Model Hierarchy with Subsystems



Model System Components

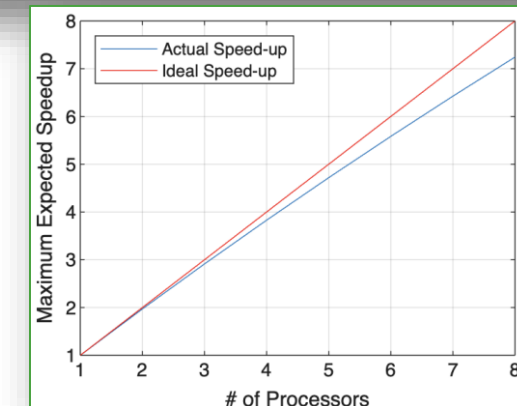
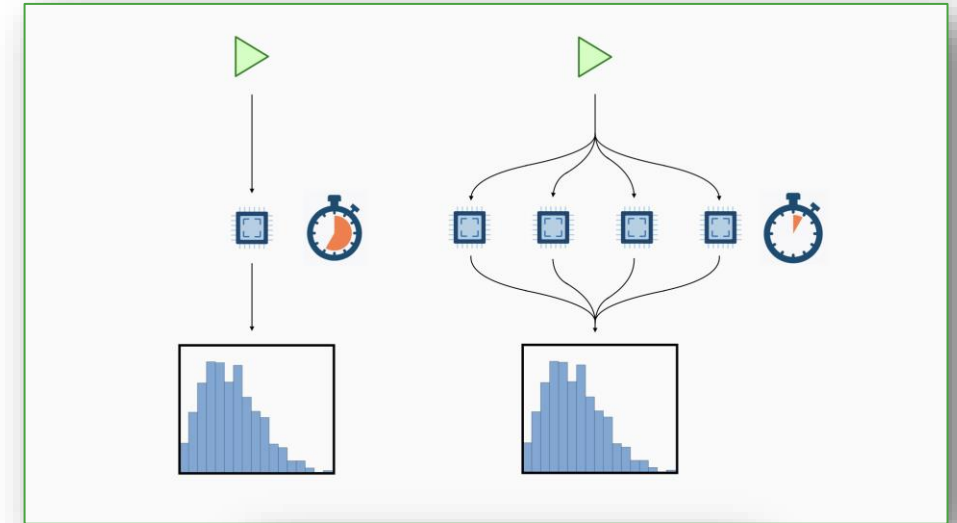


Configure Simulink Solvers to Improve Performance

Parallel Computing Onramp

- Bezplatný krátky kurz o zrýchlení kódu využitím viacerých jadier CPU
 - krátke videá a praktické cvičenia
 - automatizované hodnotenia
 - okamžitá spätná väzba

- Témy zahŕňajú:
 - konverziu sériového kódu na paralelný
 - parallel pool a MATLAB workers
 - premenné v parfor cykloch
 - prenos dát a komunikácia



Motor Control Algorithms with Simulink and Simscape

- Kurz o modelovaní a riadení PMSM s Motor Control Blockset and Simscape Electrical
 - krátke videá a praktické cvičenia
 - automatizované hodnotenia
 - okamžitá spätná väzba
- Témy zahŕňajú:
 - modelovanie elektromotorov
 - presnosť modelu
 - riadenie V/F a vektorové riadenie (FOC)
 - automatické ladenie zosilnení



Introduction to Motor Control



Motor Modeling with Simscape Electrical

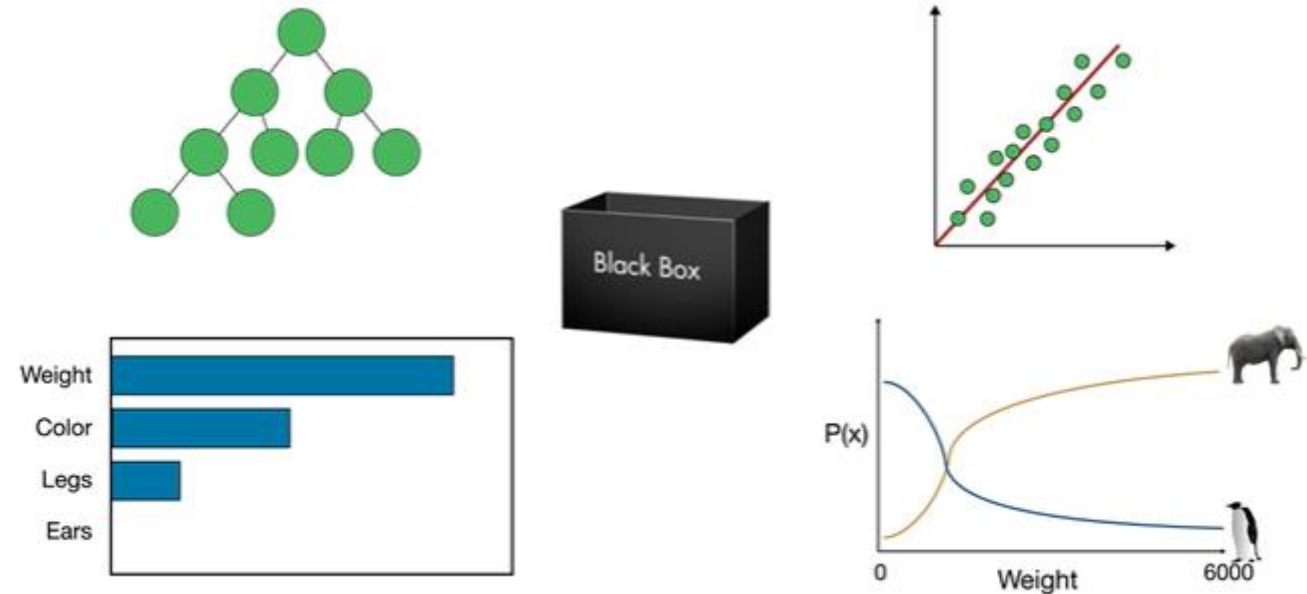


Control Techniques for PMSM

Explainable Machine Learning

- Kurz o interpretovaní rozhodnutí modelov strojového učenia
 - krátke videá a praktické cvičenia
 - automatizované hodnotenia
 - okamžitá spätná väzba

- Témy zahŕňajú:
 - interaktívne trénovanie modelu
 - interpretovateľné modely
 - globálne vysvetlenie modelu
 - lokálne vysvetlenie modelu



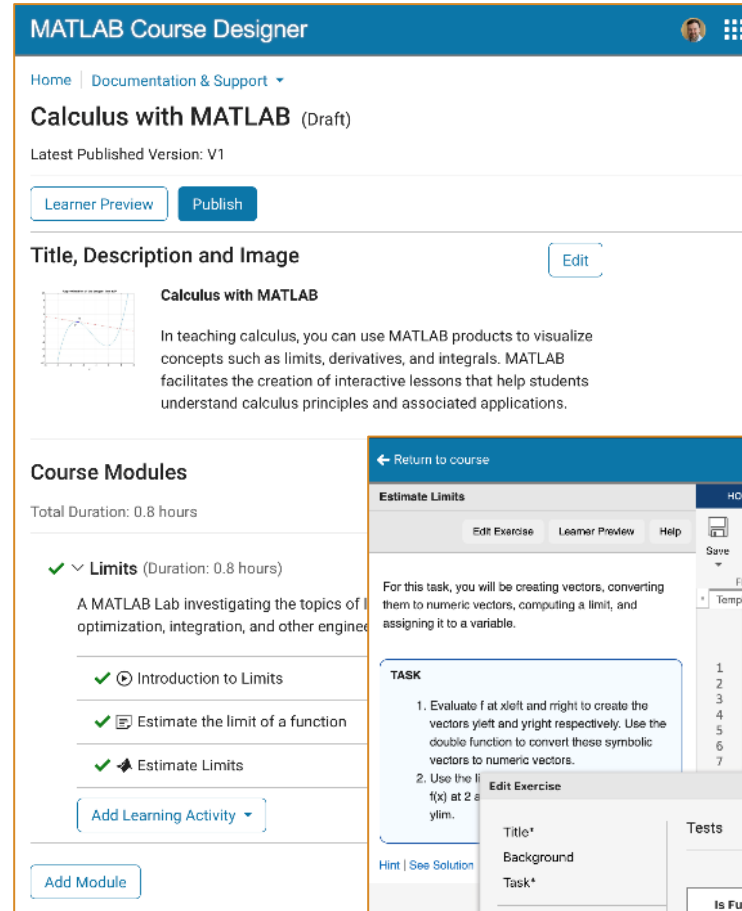
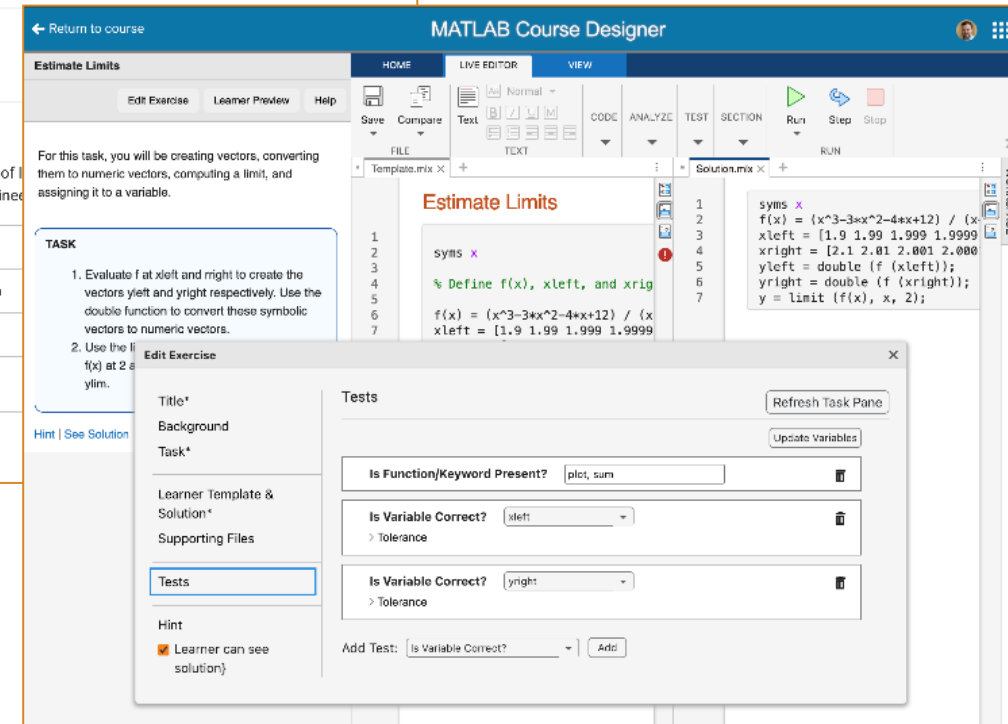
Teaching with MATLAB

- Kurz zameraný na učiteľov
 - tvorba pútavých predmetov s online nástrojmi
- Tvorba interaktívnych skriptov
 - zapojenie študentov pomocou Live skriptov
- Zdieľanie obsahu v MATLAB Drive
 - zdieľanie materiálov s kolegami a študentmi
- Podpora výučby prostredia MATLAB
 - zahrnutie vzdelávacích materiálov
- Hodnotenie študentov
 - automatické hodnotenie v MATLAB Grader
- Spolupráca s komunitou

Explore Course Lessons		
1.	Introduction Use MATLAB to increase student engagement in your course.	1 min
2.	Creating Interactive Scripts Engage your students with live scripts.	30 mins
3.	Sharing Content Use MATLAB Drive to share files with course collaborators and students.	15 mins
4.	Helping Students Learn MATLAB Incorporate MathWorks learning resources into your instruction.	10 mins
5.	Assessing Students Create and automatically grade MATLAB coding assignments with MATLAB Grader.	30 mins
6.	Engaging with the MATLAB Community Find resources and get support through the MathWorks community	5 min
7.	Conclusion Find additional resources and give feedback on the course.	1 min

MATLAB Course Designer

- Tvorba a úprava aktivít
- Zoskupenie aktivít pre spoluprácu a zdieľanie
- Hodnotenia s automatickým známkovaním
- Sledovanie priebehu a dokončenia aktivít
- Texty, videá, interakcie s MATLABom a Simulinkom
- 1EdTech certifikácia pre integráciu LMS pomocou LTI 1.3

Tvorba kurzu

Vytvorenie kurzu

Náhľad kurzu

Vytvorte aktivity na čítanie,
video, cvičenia v
MATLABe a Simulinku

MATLAB Course Designer

[Home](#) | [Documentation & Support](#)

Simulink Basics (Draft)

Learner Preview

Publish

Title, Description and Image



Simulink Basics

No description yet.

Edit

Course Modules

Total Duration: 0.0 hours

Simulink Exercise (Duration: 0.0 hours)

Edit | Learner Preview | 

✓  Course Overview Video	Edit 
✓  Reading	Edit 
✗  Getting to know MATLAB Variables	Edit 
✓  MATLAB Variables in Simulink	Edit 
✓  Blocks and Parameters	Edit 
✓  Vectorize Multiple Signals	Edit 

Add Learning Activity

Add Module

Publikovanie pre študentov

Vkladanie videí

MATLAB Course Designer

Home | Documentation & Support

Simulink **Názov**

URL

Title, Description and Image

Course Modules

Total Duration **Náhľad**

Simulink Exercise (Duration)

Course Overview Video

Reading

Getting to know MATLAB

MATLAB Variables in Simulink

Blocks and Parameters

Vectorize Multiple Signals

Add Learning Activity

Add Module

Edit Video

Title* MATLAB/Simulink Tutorial for Beginners
Characters remaining: 212

URL* <https://www.youtube.com/watch?v=vxzR3W2BcRk> **Check URL**

Supported platforms: YouTube™ and Vimeo™. Refer to the service provider for instructions for generating a shareable video URL.

MATLAB Simulink Tutorial for Beginners (Step-by-Step!)

MATLAB - Simulink Tutorial

Watch on YouTube

Cancel **Update**

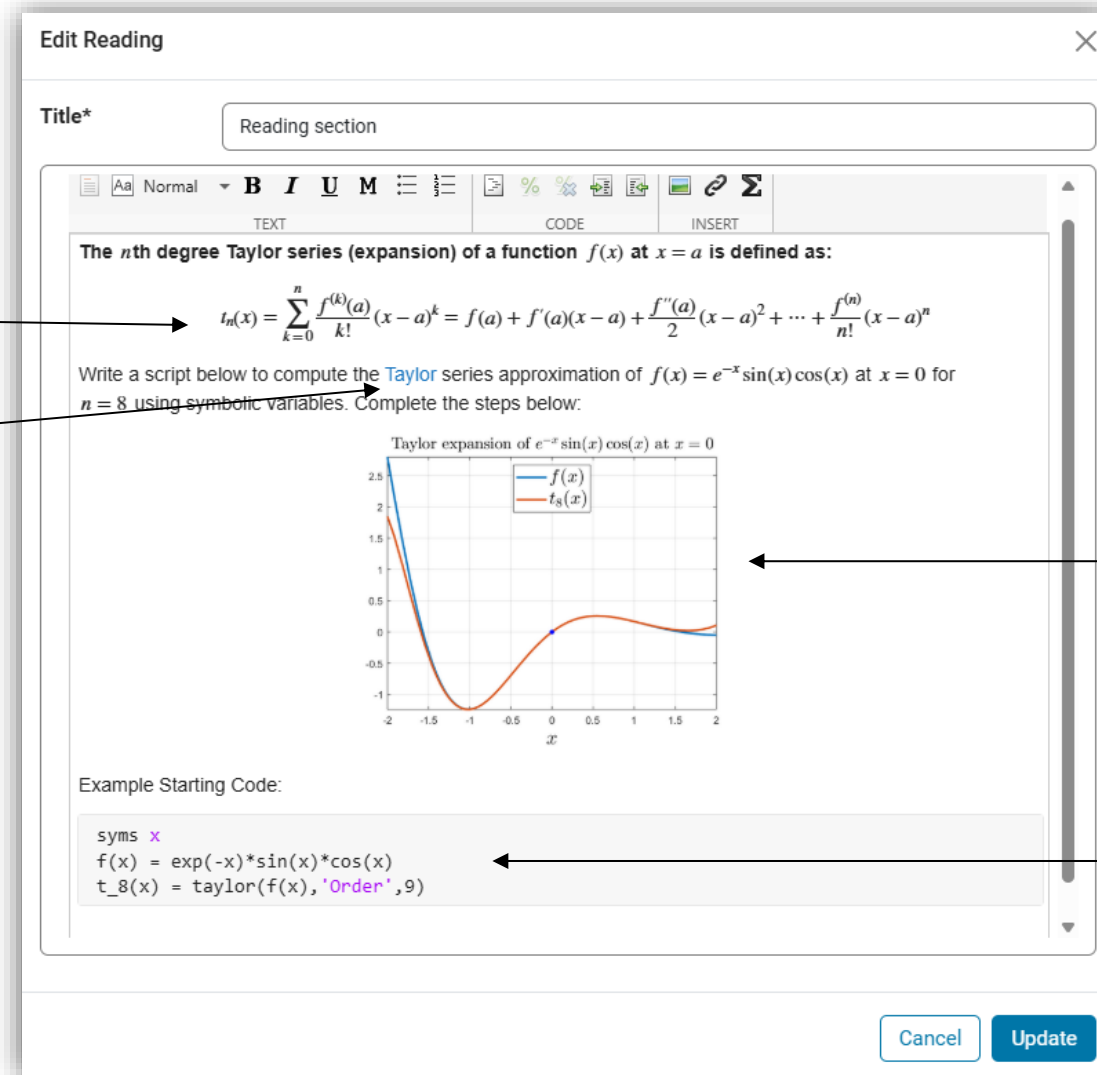


Podporované platformy

Textový popis

Latex vzorec

Odkaz



The screenshot shows a window titled "Edit Reading" with a close button (X) in the top right corner. Below the title bar is a "Title*" field containing the text "Reading section". The main editing area has a toolbar with icons for text formatting (bold, italic, underline, monospace), lists, links, and mathematical symbols. The text content includes:

- A paragraph: "The n th degree Taylor series (expansion) of a function $f(x)$ at $x = a$ is defined as:"
- A LaTeX formula:
$$t_n(x) = \sum_{k=0}^n \frac{f^{(k)}(a)}{k!} (x-a)^k = f(a) + f'(a)(x-a) + \frac{f''(a)}{2} (x-a)^2 + \dots + \frac{f^{(n)}(a)}{n!} (x-a)^n$$
- A paragraph: "Write a script below to compute the [Taylor](#) series approximation of $f(x) = e^{-x} \sin(x) \cos(x)$ at $x = 0$ for $n = 8$ using symbolic variables. Complete the steps below:"
- A plot titled "Taylor expansion of $e^{-x} \sin(x) \cos(x)$ at $x = 0$ ". The plot shows two curves: $f(x)$ (blue) and $t_8(x)$ (orange). The x-axis ranges from -2 to 2, and the y-axis ranges from -1 to 2.5. The curves are nearly identical near $x = 0$.
- A code block titled "Example Starting Code:" containing the following Python code:


```
syms x
f(x) = exp(-x)*sin(x)*cos(x)
t_8(x) = taylor(f(x), 'Order', 9)
```

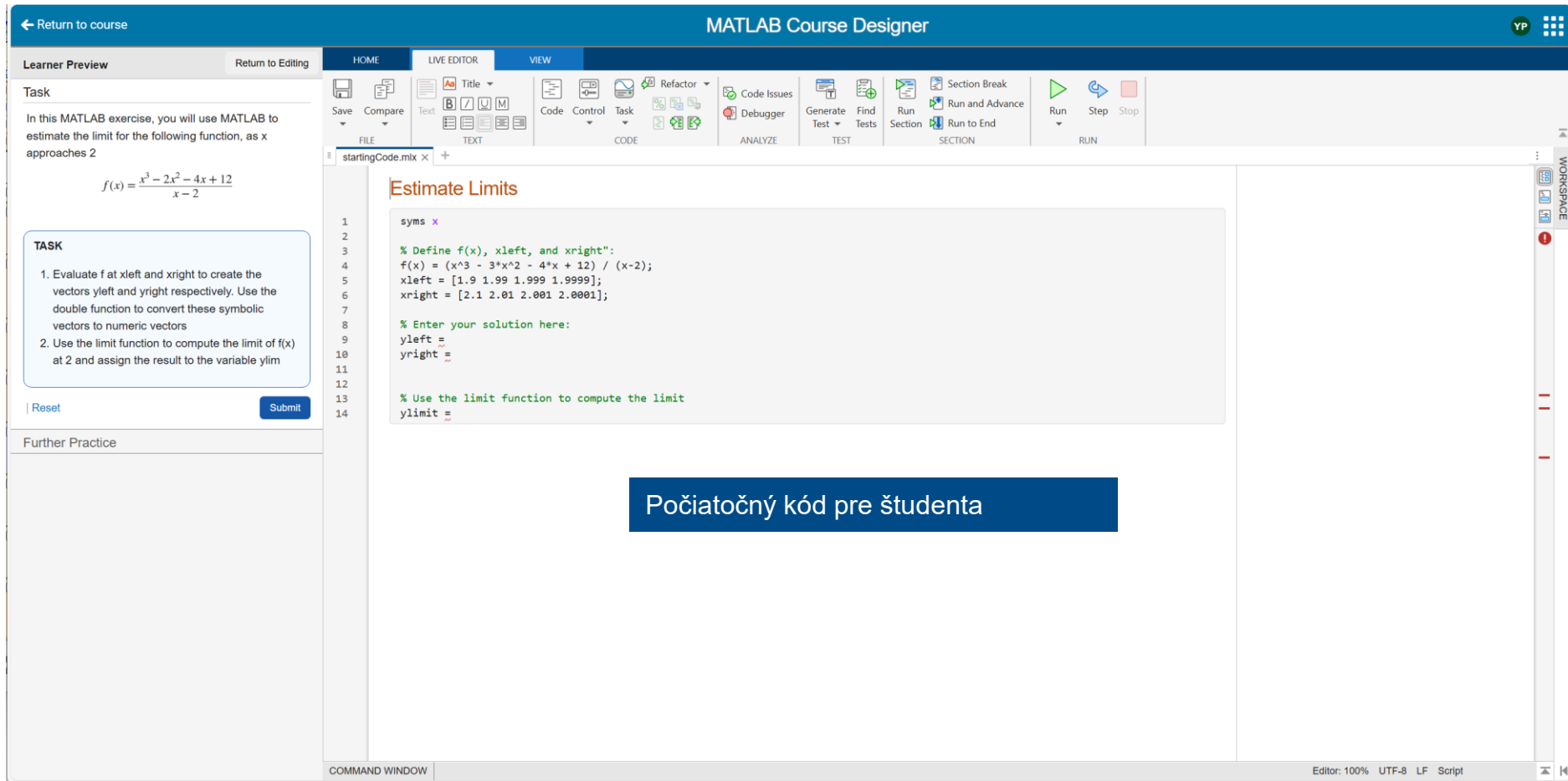
At the bottom right of the window are "Cancel" and "Update" buttons.

Obrázok

Ukážka kódu

Testy, pomocné
súbory a nápoveda

Ako uvidí úlohu študent



The screenshot shows the MATLAB Course Designer interface. On the left, the 'Learner Preview' pane displays a task description and a mathematical function. The main area shows the 'LIVE EDITOR' with a code editor titled 'Estimate Limits' containing MATLAB code for defining the function and computing limits. The bottom status bar indicates 'Editor: 100% UTF-8 LF Script'.

Task Description:

In this MATLAB exercise, you will use MATLAB to estimate the limit for the following function, as x approaches 2

$$f(x) = \frac{x^3 - 2x^2 - 4x + 12}{x - 2}$$

TASK

1. Evaluate f at x_{left} and x_{right} to create the vectors y_{left} and y_{right} respectively. Use the `double` function to convert these symbolic vectors to numeric vectors
2. Use the `limit` function to compute the limit of $f(x)$ at 2 and assign the result to the variable `y_{\text{lim}}`

Code Editor Content:

```

1 syms x
2
3 % Define f(x), xleft, and xright:
4 f(x) = (x^3 - 3*x^2 - 4*x + 12) / (x-2);
5 xleft = [1.9 1.99 1.999 1.9999];
6 xright = [2.1 2.01 2.001 2.0001];
7
8 % Enter your solution here:
9 yleft =
10 yright =
11
12
13 % Use the limit function to compute the limit
14 ylimit =
  
```

Počiatkový kód pre študenta

Úloha v Simulinku – tvorba obsahu

Return to course

Use MATLAB Variables

Learner Preview Help

Exercise Basics

Task 1

Task 2

As you saw in the overview activity, you will eventually incorporate an externally-provided nozzle command that must also be true (1) for the nozzle to extrude. For now, you will simulate this external signal using a Step block.

TASK

1. Add a **Step** block and a **Logical Operator** block to the model. Leave the default Operator, **AND** selected.
2. Connect the **Step**, **Compare to Constant**, and **Logical Operator** blocks so that the Logical Operator outputs 1 when both the **Step** and **Compare to Constant** blocks produce the value 1.
3. Connect the output of the **Logical Operator** block to the **Signal data** assessment block.

Add Task

Panel úloh

MATLAB Course Designer

Task 2

Background

Task*

Solution

Supporting Files

Assessment

Hint

* = Required

Background

Refresh Preview

Normal

B I U M

TEXT

CODE

INSERT

As you saw in the overview activity, you will eventually incorporate an externally-provided nozzle command that must also be true (1) for the nozzle to extrude. For now, you will simulate this external signal using a Step block.

Editovanie

WORKSPACE

COMMAND WINDOW

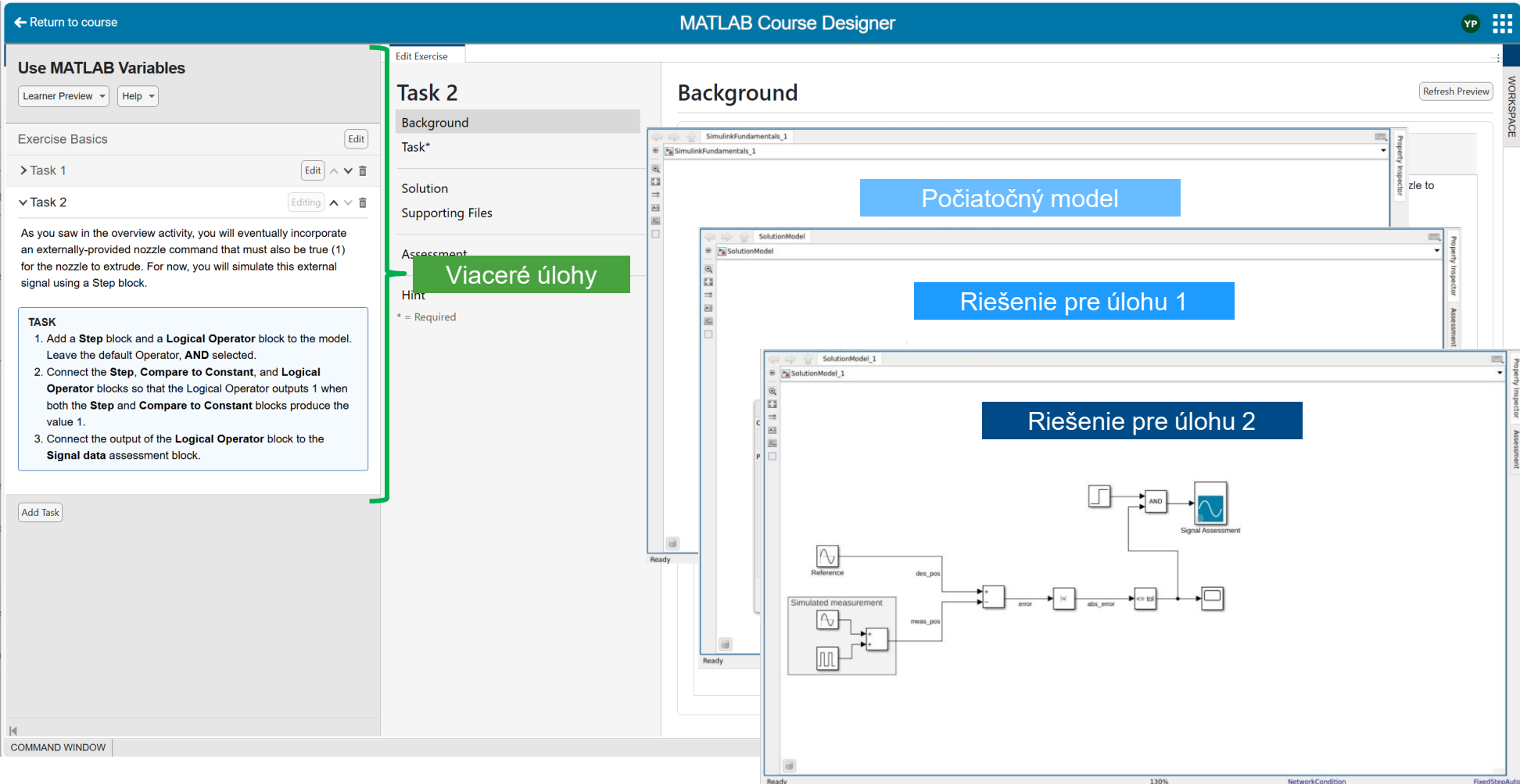
17

Úloha v Simulinku – viaceré riešenia

Počiatkový model

Riešenie pre úlohu 1

Riešenie pre úlohu 2



The screenshot shows the MATLAB Course Designer interface. On the left, a sidebar lists tasks under 'Task 2'. A green box labeled 'Viaceré úlohy' (Multiple tasks) highlights the task list. The main workspace displays three Simulink models stacked vertically, each with a blue label: 'Počiatkový model' (Initial model), 'Riešenie pre úlohu 1' (Solution for task 1), and 'Riešenie pre úlohu 2' (Solution for task 2). The bottom model shows a control system diagram with blocks for Reference, Simulated measurement, error, abs_error, and Signal Assessment.

Úloha v Simulinku – vyhodnocovanie

Return to course

MATLAB Course Designer

Use MATLAB Variables

Learner Preview Help

Exercise Basics

Task 1

Task 2

As you saw in the overview activity, you will eventually incorporate an externally-provided nozzle command that must also be true (1) for the nozzle to extrude. For now, you will simulate this external signal using a Step block.

TASK

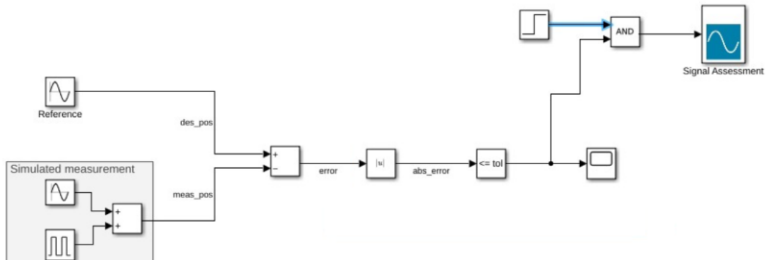
1. Add a **Step** block and a **Logical Operator** block to the model. Leave the default Operator, **AND** selected.
2. Connect the **Step**, **Compare to Constant**, and **Logical Operator** blocks so that the Logical Operator outputs 1 when both the **Step** and **Compare to Constant** blocks produce the value 1.
3. Connect the output of the **Logical Operator** block to the **Signal data** assessment block.

Add Task

Edit Exercise Solution x

SolutionModel_1

Stop Time 10.0



Ready

109%

Assessment

Add tests based on your solution. To test elements such as blocks and connections, select the element and then add the test.

Add Test

- Block existence
- Block parameter value
- Block connection
- Model parameter value
- Signal data
- Custom

Signal data

Connect a signal

Capture Signal

Signal data capture

Signal source: Logical Operator:2

Tolerance 0

Block connection

Source

Destination

Add Destination

Block connection

Source

Destination

Add Destination

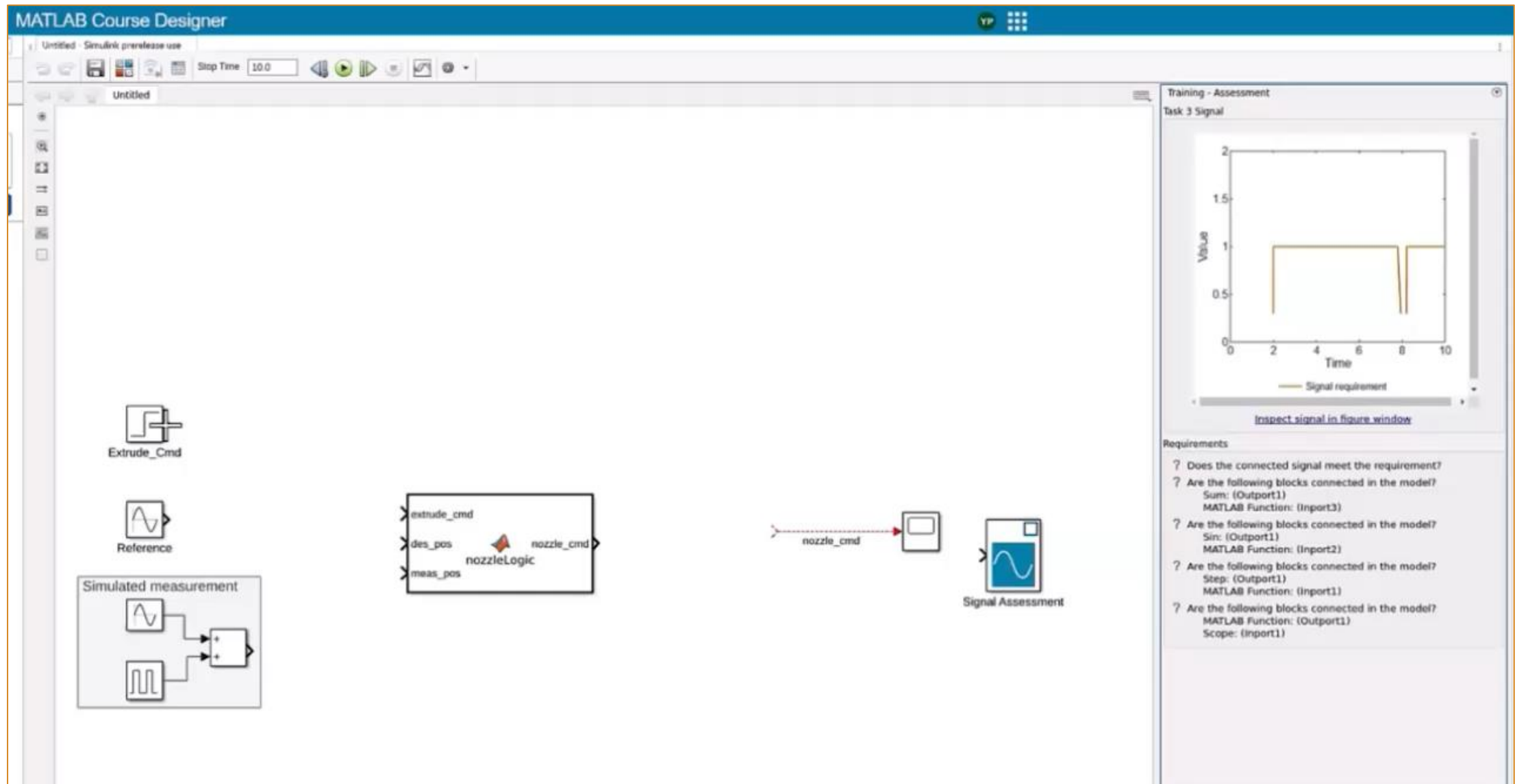
NetworkCondition

auto(FixedStepDiscrete)

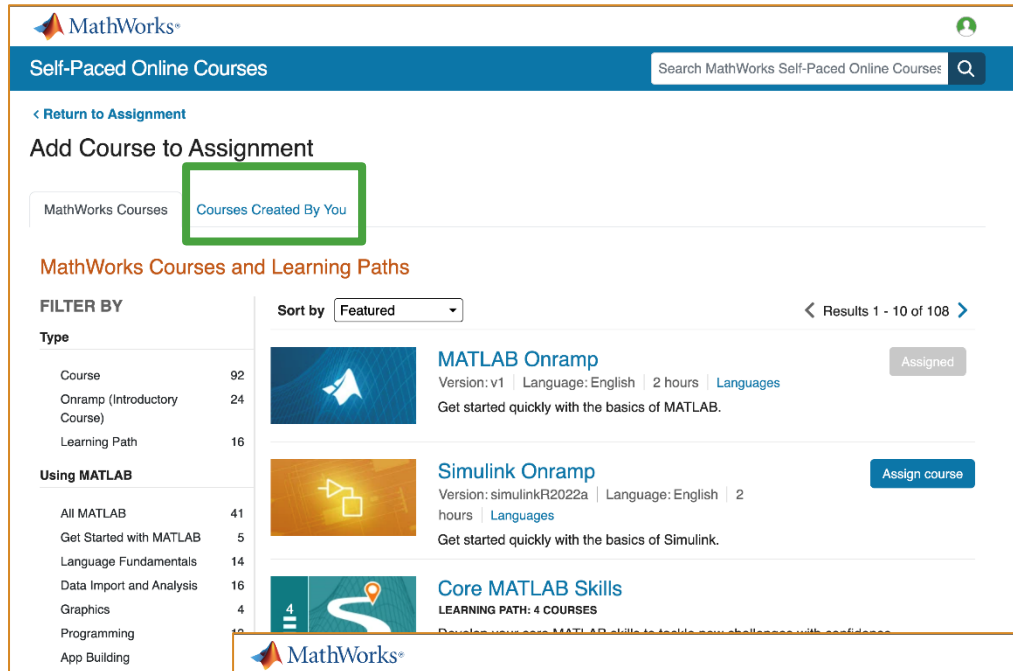
Zjednodušený Simulink Editor

Panel
vyhodnotenia

Posúdenie modelu v prostredí Simulink



Integrácia s LMS



MathWorks®

Self-Paced Online Courses

Search MathWorks Self-Paced Online Courses

< Return to Assignment

Add Course to Assignment

MathWorks Courses Courses Created By You

MathWorks Courses and Learning Paths

FILTER BY

Type

Course	92
Onramp (Introductory Course)	24
Learning Path	16

Using MATLAB

All MATLAB	41
Get Started with MATLAB	5
Language Fundamentals	14
Data Import and Analysis	16
Graphics	4
Programming	16
App Building	16

Sort by Featured

Results 1 - 10 of 108

MATLAB Onramp

Version: v1 | Language: English | 2 hours | Languages

Get started quickly with the basics of MATLAB.

Assigned

Simulink Onramp

Version: simulinkR2022a | Language: English | 2 hours | Languages

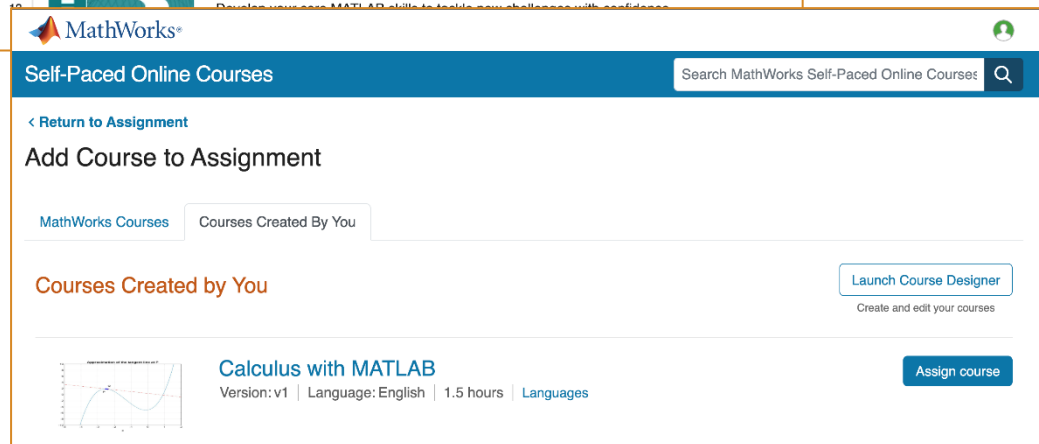
Get started quickly with the basics of Simulink.

Assign course

Core MATLAB Skills

LEARNING PATH: 4 COURSES

Develop your core MATLAB skills to tackle new challenges with confidence.



MathWorks®

Self-Paced Online Courses

Search MathWorks Self-Paced Online Courses

< Return to Assignment

Add Course to Assignment

MathWorks Courses Courses Created By You

Courses Created by You

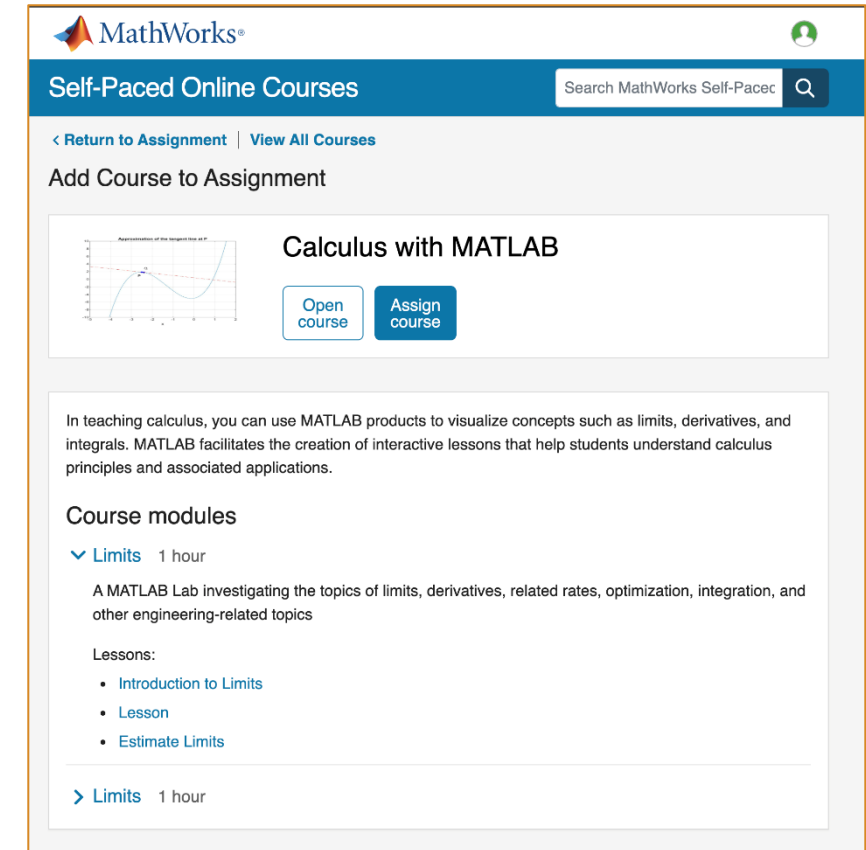
Launch Course Designer

Create and edit your courses

Calculus with MATLAB

Version: v1 | Language: English | 1.5 hours | Languages

Assign course



MathWorks®

Self-Paced Online Courses

Search MathWorks Self-Paced Online Courses

< Return to Assignment | View All Courses

Add Course to Assignment

Calculus with MATLAB

Open course Assign course

In teaching calculus, you can use MATLAB products to visualize concepts such as limits, derivatives, and integrals. MATLAB facilitates the creation of interactive lessons that help students understand calculus principles and associated applications.

Course modules

> Limits 1 hour

A MATLAB Lab investigating the topics of limits, derivatives, related rates, optimization, integration, and other engineering-related topics

Lessons:

- Introduction to Limits
- Lesson
- Estimate Limits

> Limits 1 hour

Učiteľ priraduje študentom kurz prostredníctvom existujúcej aplikácie Online Courses App v LMS

Dokumentáció


MATLAB Help Center

Resources

Teaching and Learning

Build engineering and science skills using MathWorks® products

Teaching and learning tools at MathWorks are designed to support individual faculty teaching with MATLAB® and Simulink® in universities, colleges, and community colleges.

When your goal is to teach engineering and science skills using MathWorks products, our teaching and learning applications can help you achieve your objectives in any learning environment, including in-classroom, virtual, and hybrid programs.

With these applications, you can:

- Track learner progress and completion.
- Create interactive course assignments.
- Assess learner skills automatically.

Products for Teaching and Learning

MATLAB Course Designer

Create and assign courses, courseware, labs, and assessments

MATLAB Grader

Automatically grade MATLAB code in any learning environment

MATLAB Onramp

Learn MATLAB

Topics

Track Learner Progress in Online Courses

Invite Learners to Share Progress (MATLAB and Simulink Online Courses)
Monitor course completion by inviting learners to share their course progress and certificate.

Track Learner Progress (MATLAB and Simulink Online Courses)
Use the shared progress page to see how your learners are progressing.

Create Assessment Items for Skills Evaluation
Test Solutions from Learners (MATLAB Grader)
Create assessment tests that check for various common errors or conditions.

MATLAB Course Designer

Create and assign courses, courseware, labs, and assessments

Use MATLAB® Course Designer™ to create your own self-paced online courses, similar to MATLAB Onramp. Add learning activities such as text, videos, and interactive exercises to your course. Then, assign your course to learners from your learning management system (LMS) and track their progress in the same way you would with courses authored by MathWorks®.

Topics

Course Setup

Create Courses Using MATLAB Course Designer
Create your own self-paced online courses.

Update Courses in MATLAB Course Designer
Make changes to assigned courses.

MATLAB Exercises

Create MATLAB Exercise Using MATLAB Course Designer
Create a MATLAB exercise and include it in your self-paced online course.

MATLAB Exercise Assessment Tests in MATLAB Course Designer
Test learner solutions to MATLAB exercises in MATLAB Course Designer.

Simulink Exercises

Create Simulink Exercise Using MATLAB Course Designer
Create a Simulink® exercise and include it in your own self-paced online course.

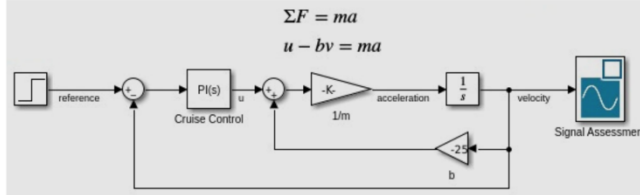
Simulink Exercise Assessment Tests in MATLAB Course Designer
Test learner solutions to Simulink exercises in MATLAB Course Designer.


MATLAB Help Center

Add Task Solution

The solution for a task in a Simulink exercise is a Simulink model that represents the correct answer for the task. The solution lets you assess the learner's solution for the task. To provide further practice without an assessment, you can leave the solution empty.

For example, the solution for a block diagram with two unconnected signal lines can be a completed block diagram with those signal lines connected.



To add a solution:

- On the **Edit Task** tab, in the left navigation pane, click **Solution**.
- Choose how to add the solution.
 - Copy** – Copy the starting model or a solution from another task.
 - New** – Create a blank model.
 - Upload** – Upload a model from your file system.

Solution

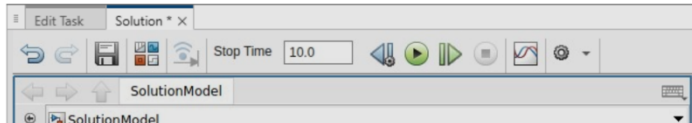
SolutionModel.slx Open Copy New Upload

To modify the solution:

- Next to the solution filename, click **Open**.
- On the **Solution** tab, modify the solution. If you copy the starting model or the model from the previous task, update the model the same way you expect learners to update the model for this task.

MATLAB Course Designer saves changes automatically. If you close the browser tab or sign out while editing the solution, your recent changes might not be saved. When you modify an uploaded model, your changes are saved to the exercise, not to the original model in your file system.

Simulink exercises in MATLAB Course Designer use a simplified toolbar that provides a subset of common actions, such as specifying the simulation stop time, simulating the model, and viewing logged data. To access other actions, click the canvas and press **Ctrl+Period (.)**. Then, search for an action.



Teaching and Learning

MATLAB Course Designer

Authoring Workflows: MATLAB & Simulink

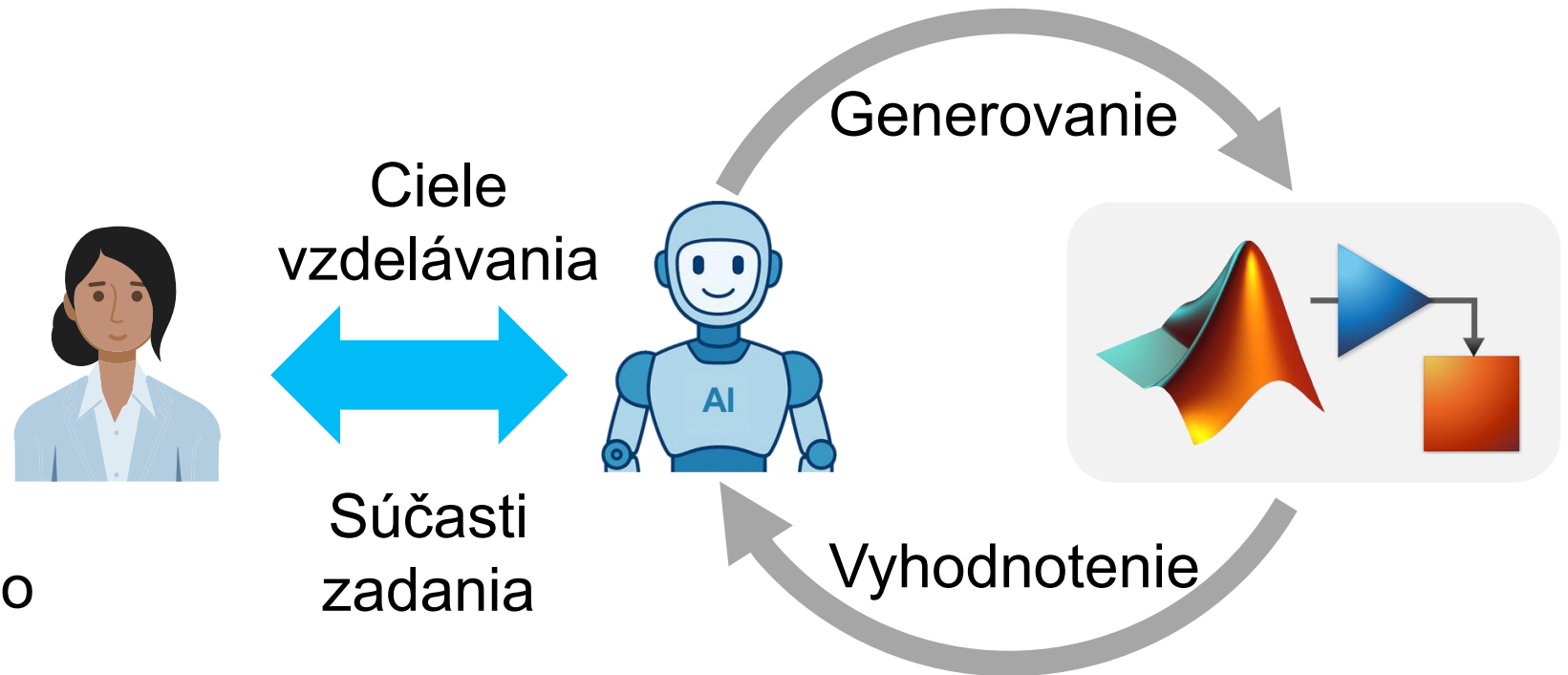
Vytvorenie zadania s testovaním

Úloha:

Ako môžem pomôcť učiteľom s tvorbou zadaní?

Riešenie:

Zapojte odborné znalosti do návrhu vášho hodnotenia pomocou Agentic AI.



Cieľ: Vytvorte celú sadu zadaní s hodnotením v pripravených na import do nástroja MATLAB Course Designer alebo MATLAB Grader.

Ukážka: Vytvorenie zadania

```
codex exec demos/assessment-generation-for-matlab-grader

**MATLAB Grader Assessment Item Generation Complete**

Selected item: `Vector Transpose Function Item`
Purpose: Formative
Type: Function assessment item
Chosen for novice learners because it directly checks row
single transpose operation without adding unrelated MATLAB

Generated files in codex-demo-output/vector_transpose_fun

- description.txt
- solution.m
- template.m
- function_call.m
- tests.m

Candidate ideas considered: `Identify
Column Forms` (Medium), `Normalize Mix
Easy candidate for the novice introduc

Verification: MATLAB executed the gene
successfully.
```

```
matlab_grader_vector_transpose_review.m x +
/Users/jalderso/blogs/matlab_grader_vector_transpose_review.m

description.txt
Vector Transpose Function Item

Write a MATLAB function that identifies whether an input vector is a row vector or a column vector, then uses a transpose
operation to convert it to the opposite orientation. This formative assessment item focuses on recognizing vector shape and
using transpose syntax to move between row and column forms.

Instructions:
1. Create a function named vector_transpose_function_item that accepts one numeric vector v.
2. Determine whether v is a row vector or a column vector.
3. Return orientation as the character vector 'row' when v is a row vector, or 'column' when v is a column vector.
4. Return convertedVector as v transposed to the opposite orientation.

Required function signature:

function [orientation, convertedVector] = vector_transpose_function_item(v)

Hints:
- isrow and iscolumn can help you inspect the orientation of a vector.
- The transpose operator changes a row vector into a column vector and a column vector into a row vector.
- You may assume v is a nonempty row or column vector with more than one element.

Before you submit:
```

```
codex exec demos/assessment-generation-for-matlab-grader

Verification: MATLAB executed the generated tests against the reference solution
successfully.

%% Test 1: identify a random row vector
v = randi([1, 30], 1, 6);
[orientation, ~] = vector_transpose_function_item(v);
assessVariableEqual('orientation', 'row');

%% Test 2: transpose a row vector into a column vector
v = randperm(19, 7) - 10;
[~, convertedVector] = vector_transpose_function_item(v);
assessVariableEqual('convertedVector', v);
```

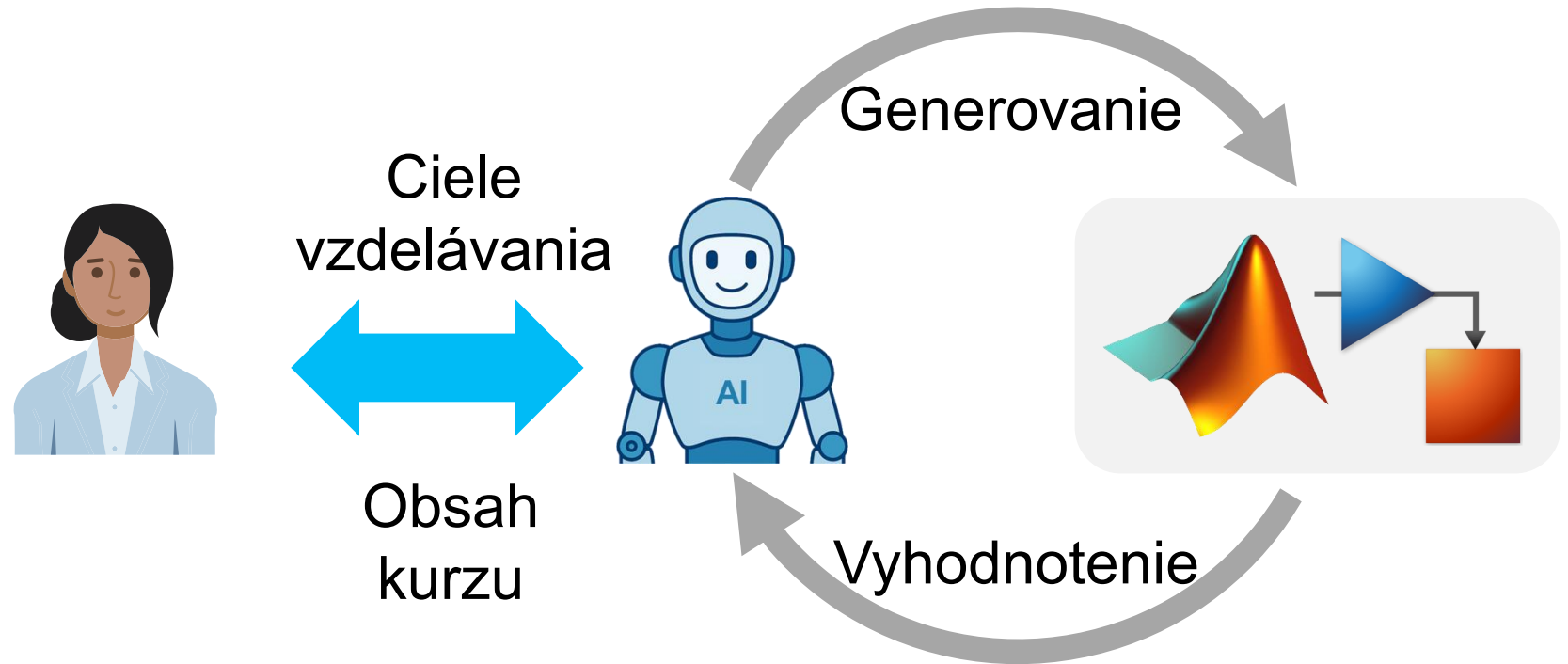
Vytvorenie kurzu

Úloha:

Zosúladienie vzdelávacích materiálov a hodnotenia

Riešenie:

Aplikácia pravidiel cieľov, postupov, usmernené cvičenia, hodnotenie a dôkladná kontrola.



Cieľ: Vytvorte si jednu lekciu alebo celý kurz s obsahom a hodnoteniami, pripravené na import do vášho LMS alebo MATLAB Course Designera

Vytvorenie kurzu

Complete output

Module

- Course

- Audience

- Delivery

- Activities

- Scope

General

- course

- instructions

- export

- MATLAB

- Course

Vector Orientation in MATLAB

Course Purpose

This compact module introduces students to MATLAB, where they can distinguish row and column vectors and transpose operations.

Audience

First-year engineering students in a MATLAB course.

Prerequisites

- Basic MATLAB commands
- Numeric arrays using row and column separators
- Awareness that semicolons end lines of code

Required Product

- MATLAB

Module

- 1. Identify and Transpose Vectors
- Reading: Vector Orientation
- Practice: Predict Shapes Based on Dimensions
- MATLAB Exercise: Vector Orientation

Persistence Note

Common Cartridge export has been added to the target LMS. Exercise instructions, and in-classroom activities for students.

Reading: Vector Orientation in MATLAB

MATLAB stores vectors as arrays. A row vector has size 1-by-n, such as `[3; 5; 8]`. The number of elements changes how the vector can be used.

Use `isrow(v)` to check whether `v` is a row vector. Use `iscol(v)` to check whether `v` is a column vector.

MATLAB has two transpose operators: `'` (conjugate transpose) and `.'` (non-conjugate transpose). The conjugate transpose changes rows into columns and complex signs. The non-conjugate transpose changes rows into columns without changing complex signs. In this introductory module, we focus on the non-conjugate transpose.

Examples:

```
rowVec = [2 4 6];
colVec = [2; 4; 6];
```

```
isrow(rowVec)
iscolumn(colVec)
```

```
colFromRow = rowVec.';
rowFromCol = colVec.';
```

After the transpose operation, `rowFromCol` is a row vector and `colFromRow` is a column vector.

Practice: Running MATLAB

For each line, predict the output. Then run the code in MATLAB to verify your prediction. Use `isrow`, and `iscolumn` to check the orientation of the vectors.

```
a = [10 20 30 40];
b = [10; 20; 30; 40];
c = a.';
d = b.';
e = c.';
```

Record the size of each vector in the table below.

Variable	Size
a	
b	
c	
d	
e	

If a prediction is wrong, check the MATLAB documentation for the `size` function or the table below.

Module 01 Activity 03: Vector Orientation and Transpose

Course Designer Placement

- Module: Identify and Transpose MATLAB Vectors
- Learning activity order: 3
- Learning activity type: MATLAB Exercise

Objective Alignment

- Course objective: Students will be able to identify row and column vectors in MATLAB and use transpose operations to convert between them.
- Module objective: Distinguish row and column vector orientation and use `'` to convert between orientations.

Learner Task

Students complete a MATLAB script that identifies row and column vectors, creates transposed copies, and applies the same operations to a second randomized data set.

MATLAB Exercise Component Source

The shared MATLAB Grader-style component files are in `grader-items/module-01-activity-03-vector-orientation-and-transpose/`.

Course Designer Authoring Fields

- Background file: `background.m`
- Task file: `task.m`
- Estimated time: 20 minutes
- Prerequisite context: MATLAB script editing, numeric arrays, row separators, semicolons
- Learner setup notes: MATLAB only; no additional toolboxes

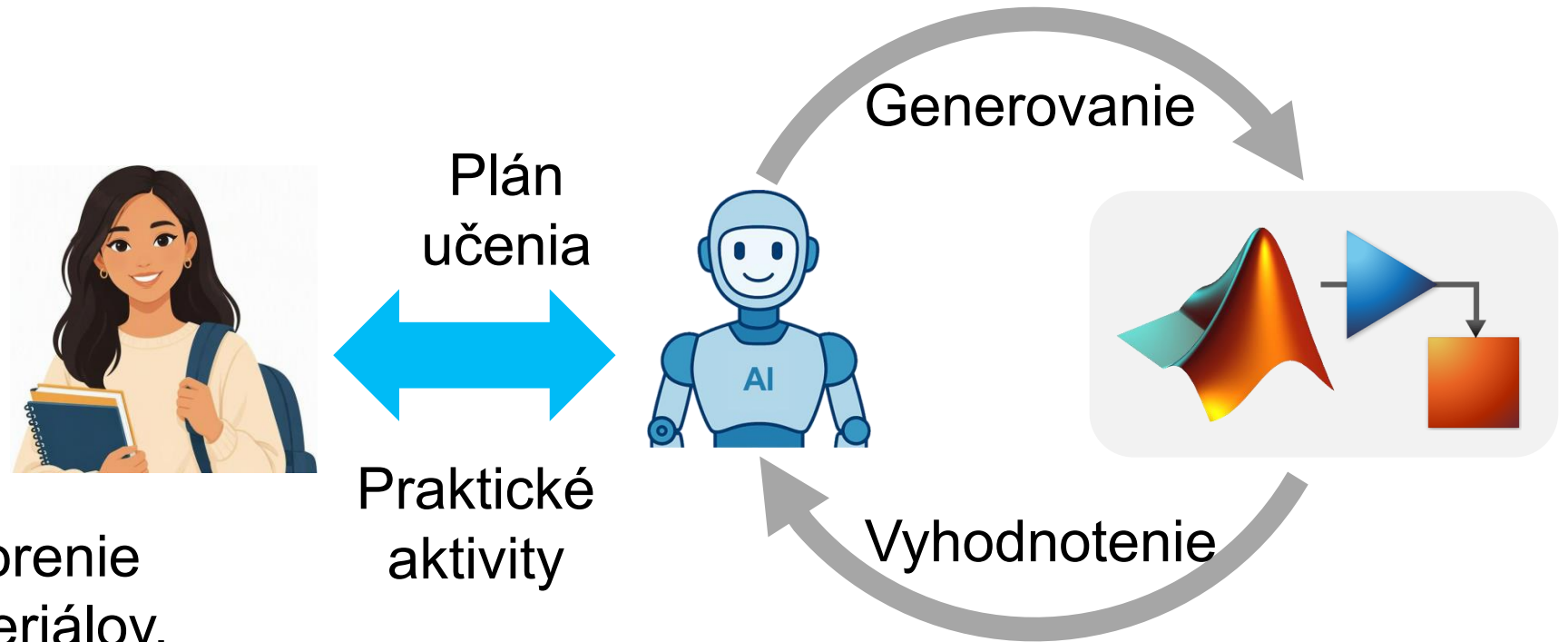
AI učiteľ pre prostredia MATLAB a Simulink

Problem:

Ako si môžem samostatne precvičiť zručnosti?

Solution:

Použijete agenta na vytvorenie personalizovaných materiálov, ktoré sú zosúladené so vzdelávacím plánom.



Cieľ: Vytvorte interaktívny vzdelávací plán s otázkami a viacerými odpoveďami, programovaním a tvorbou modelov – doplnený o hodnotenie a spätnú väzbu.

AI učiteľ pre MATLAB

codex exec demos/ai-tutoring

Learner:

I'm stuck on MATLAB vectors. I do not understand why `[1 2 3]'` and `[1; 2; 3]` seem related.

Can you help me learn how to transpose vectors without just giving me homework answers?

Tutor:

Yes. I can help with the concept and with checking you to graded homework, I won't write the final submission for you, but I can inspect your own attempt.

First diagnostic question: in MATLAB, what do you think here:
the numbers, the shape, or both?

Learner:

I think it changes the shape, but I don't know exactly

Tutor:

Good hypothesis. Let's collect MATLAB evidence instead

```
x = [1 2 3];  
y = x';  
z = [1; 2; 3];
```

```
size(x)  
size(y)  
size(z)  
isequal(y,z)
```

Learner:

MATLAB says:

```
size(x) = 1      3  
size(y) = 3      1  
size(z) = 3      1  
isequal(y,z) = 1
```

Tutor:

That evidence supports your hypothesis.

`[1 2 3]'` is a row vector: 1 row, 3 columns.

The transpose operator `'` turns it into a column vector: 3 rows, 1 column.

`[1; 2; 3]` writes the same column vector directly, because semicolons separate rows.

So these two expressions produce the same shape and values:

```
[1 2 3]'  
[1; 2; 3]
```

The MATLAB rule is: spaces or commas move across columns; semicolons move down rows; transpose swaps rows and columns.

AI učiteľ pre Simulink

codex exec demos/ai-tutoring

Learner: I am in Simulink and I do not know how to set a block parameter.
Can you make a hands-on exercise?

Tutor: Goal: practice changing a block parameter and verifying the result from the model, not from guesswork.

I opened Tutor_BlockParameterExercise.slx. The model contains Input_Value_4 -> Set_Me_Gain -> Observed_Output, with Target_Output_12 beside it.

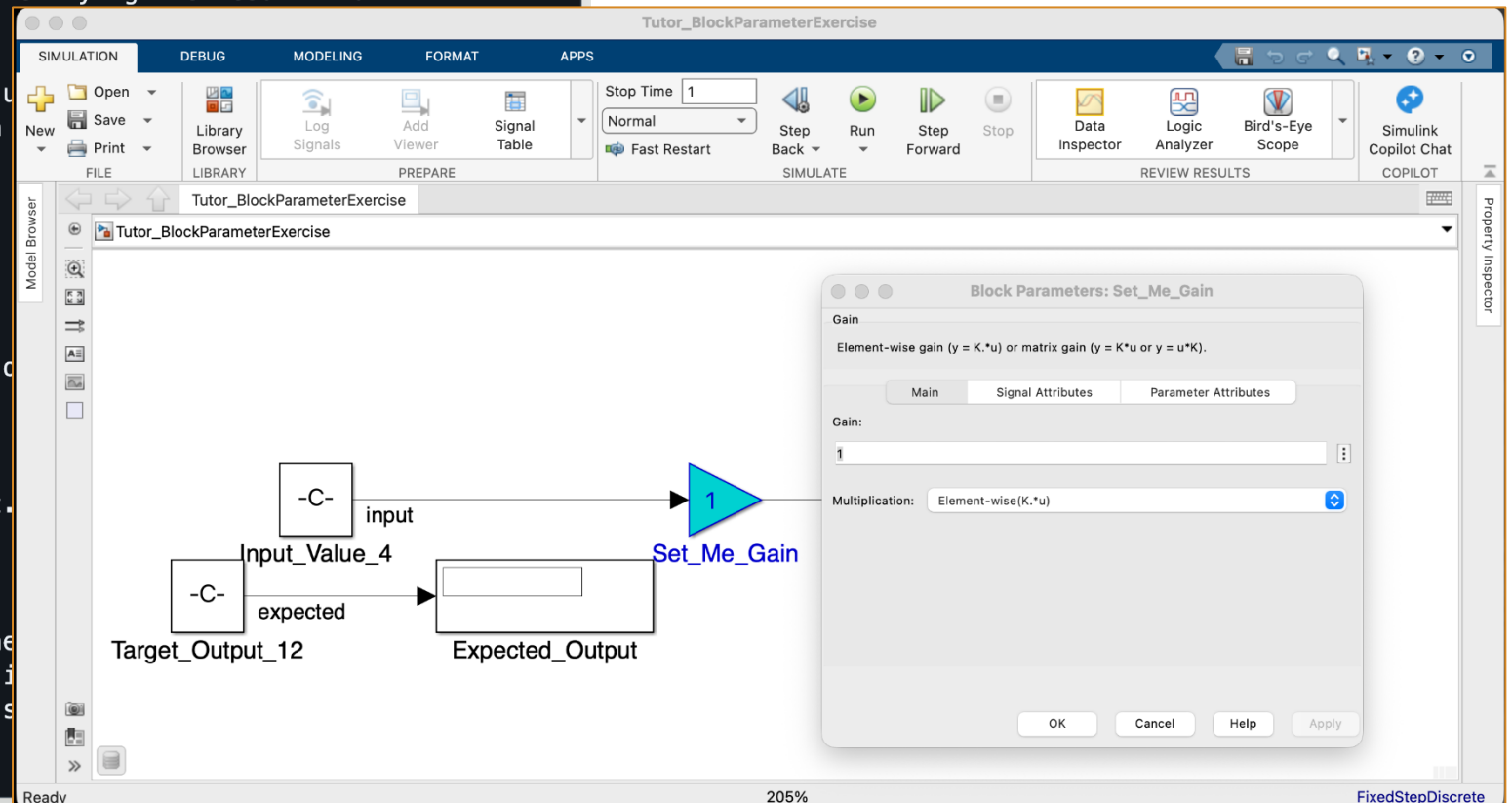
Expected evidence before you start:

- The input value is 4.
- The target gain is 3.
- The expected output is 12.
- The model starts with Set_Me_Gain.Gain = 1, so the observed output is 4.

Hands-on task:

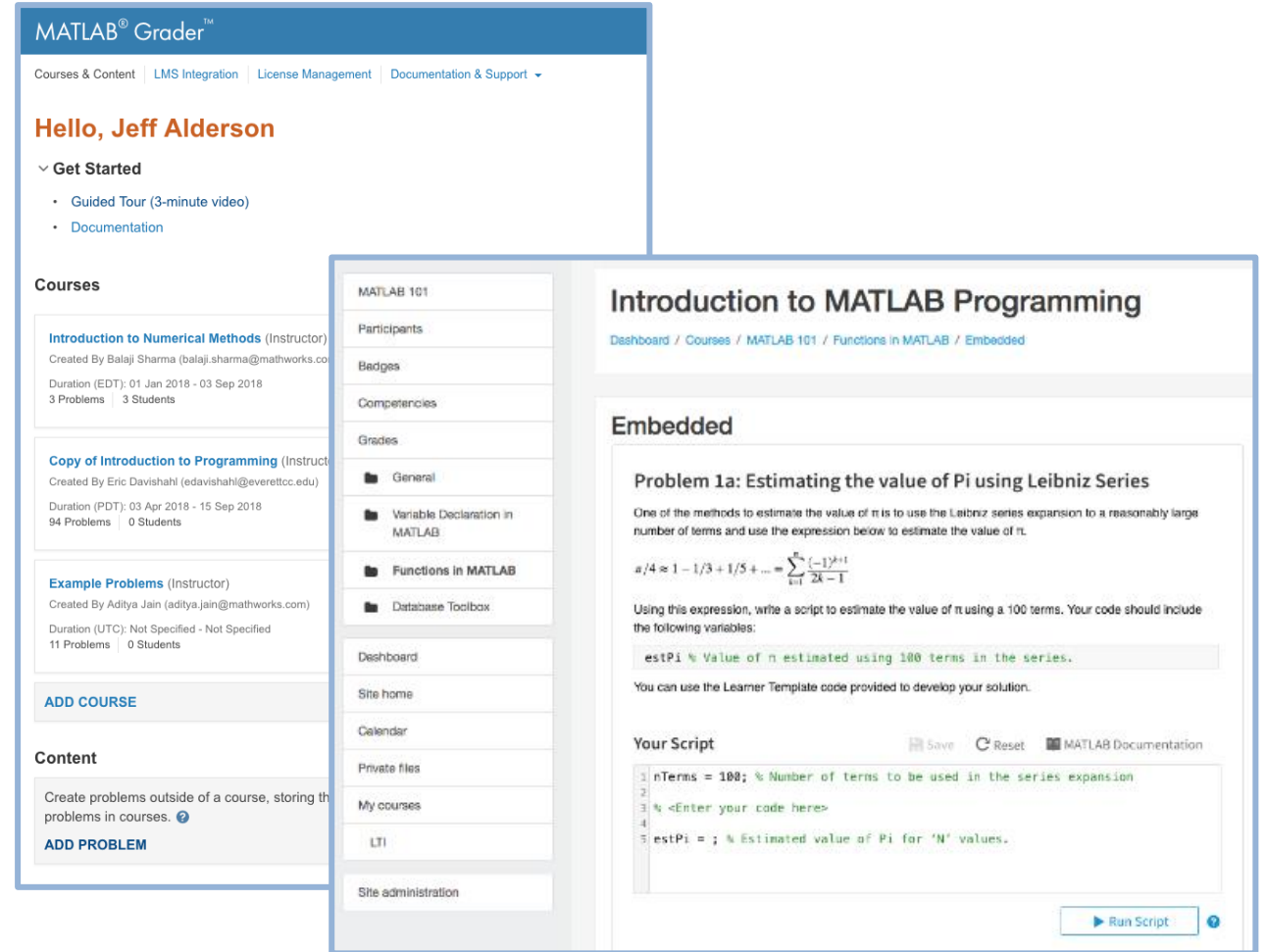
1. Double-click the Set_Me_Gain block.
2. Change the Gain parameter from 1 to K_Tutor_Target.
3. Run the model.
4. Confirm Observed_Output matches Expected_Output.

Tutor: I also created Tutor_BlockParameterExercise_check.slx to run the check. A pass means the Gain block parameter is set to the target value. A revision message means the parameter is still at the non-target value.



MATLAB Grader

- Interaktívna výuka
 - tvorba interaktívnych úloh a ich distribúcia študentom
 - zdieľanie úloh s ďalšími inštruktormi
- Automatizované hodnotenie úloh
 - okamžitá spätná väzba
 - analýza úspešnosti (študenta, skupiny)
- Možnosť integrácie s Moodle
 - alebo iným LMS
- Pre učiteľov sú pripravené úlohy



The screenshot displays the MATLAB Grader web interface. At the top, the header includes 'MATLAB® Grader™' and navigation links for 'Courses & Content', 'LMS Integration', 'License Management', and 'Documentation & Support'. A personalized greeting 'Hello, Jeff Alderson' is shown, followed by a 'Get Started' section with links to a 'Guided Tour (3-minute video)' and 'Documentation'.

The main content area is divided into two columns. The left column, titled 'Courses', lists several courses: 'Introduction to Numerical Methods' (created by Balaji Sharma), 'Copy of Introduction to Programming' (created by Eric Davisahli), and 'Example Problems' (created by Aditya Jain). Each course entry shows its duration and the number of problems and students. Below the course list are buttons for 'ADD COURSE' and 'ADD PROBLEM'.

The right column displays the details of a selected course, 'Introduction to MATLAB Programming'. It includes a breadcrumb trail: 'Dashboard / Courses / MATLAB 101 / Functions in MATLAB / Embedded'. The 'Embedded' section shows 'Problem 1a: Estimating the value of Pi using Leibniz Series'. The problem description explains the Leibniz series expansion for pi and provides the formula:
$$\pi/4 \approx 1 - 1/3 + 1/5 - \dots = \sum_{k=1}^n \frac{(-1)^{k+1}}{2k-1}$$
. It asks the user to write a script to estimate pi using 100 terms. Below the problem description is a 'Your Script' area with a text editor containing MATLAB code:

```
1 nTerms = 100; % Number of terms to be used in the series expansion
2
3 % <Enter your code here>
4
5 estPi = ; % Estimated value of Pi for 'N' values.
```

 At the bottom right of the script area is a 'Run Script' button.

MATLAB Grader – pripravené úlohy

Getting Started with MATLAB Grader 13 problems	Calculus I Created By: MathWorks 10 problems	Statistics Created By: MathWorks 15 problems
Electric Circuits Created By: MathWorks 16 problems	Digital Signal Processing Created By: MathWorks 10 problems	System Dynamics and Control Created By: MathWorks 10 problems
Dynamics Created By: MathWorks 10 problems	Introduction to Programming Created By: Eric Davishahl 111 problems	Calculus II Created By: MathWorks 10 problems
Numerical Methods Created By: MathWorks 10 problems	Symbolic Math Toolbox Created By: MathWorks 10 problems	



The screenshot shows the MATLAB Grader interface for a problem titled "Using DFT to compute convolutions". The problem description explains that the DFT is often used to compute time-domain convolutions, but users must distinguish between linear and circular convolution. It asks the user to write a function `dfcconv` that accepts three inputs: `h` (the first signal), `x` (the second signal), and `N` (an integer greater than or equal to the length of `h` and `x`). The function should return two outputs: `conv_circ` (the circular convolution) and `conv_lin` (the linear convolution). The interface includes a "Code" section with a MATLAB function template and a "Test" section with two test cases.

```
function [conv_circ, conv_lin] = dfcconv(h,x,N);
1
2
3 lx = length(x);
4 lh = length(h);
5
6 if Nmax([lx lh])
7     error('N must be at least the length of the longer signal.')
8 end
9
10 % Compute order-N circular convolution based on length-N FFTs
11
12 conv_circ = ifft(fft(h,N).*fft(x,N));
13
14 % Compute linear convolution based on appropriate-length FFTs/zero padding
15
16 M = lx+lh-1;
17 conv_lin = ifft(fft(h,M).*fft(x,M));
```

How to call the function (when the learner clicks 'Run')

```
1 h = [1 2 3];
2 x = [1 2 2 1];
3 [conv_circ,conv_lin] = dfcconv(h,x,4);
```

Assessment

- Test 1: Are the outputs correct for the test inputs? (Pretest)
 - MATLAB Code
- Test 2: Are the outputs correct for randomized inputs?
 - MATLAB Code

Pre prístup k pripraveným úlohám nám napíšte na studnicka@humusoft.cz

Podpora vzdelávacieho procesu

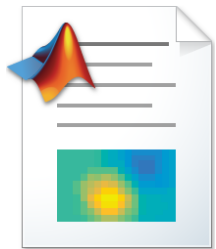
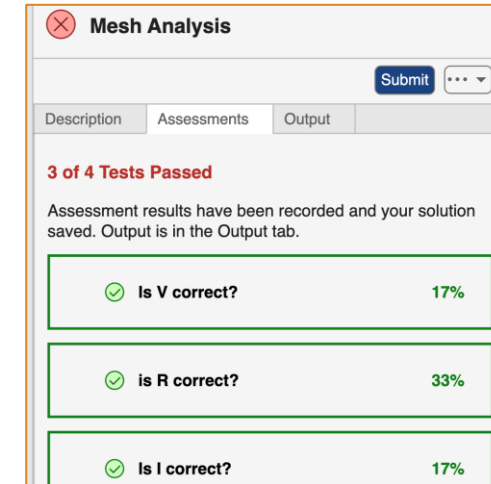
Vzdelávacie
materiály



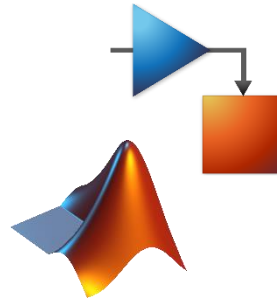
Praktické
projekty



Automatické
hodnotenie



Self-Paced
Online
Courses



**MATLAB
Course
Designer**



**MATLAB
Grader**

Ďakujem za pozornosť