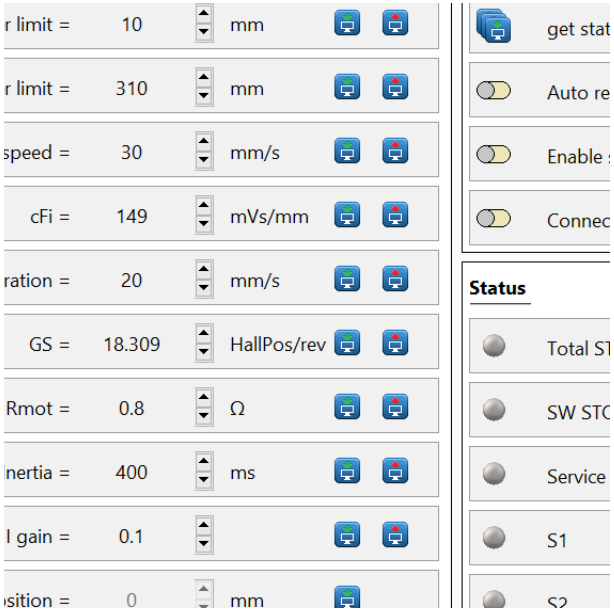
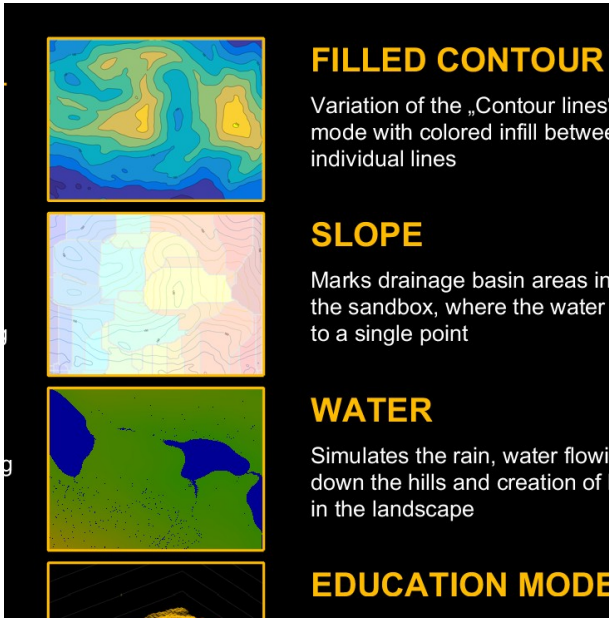
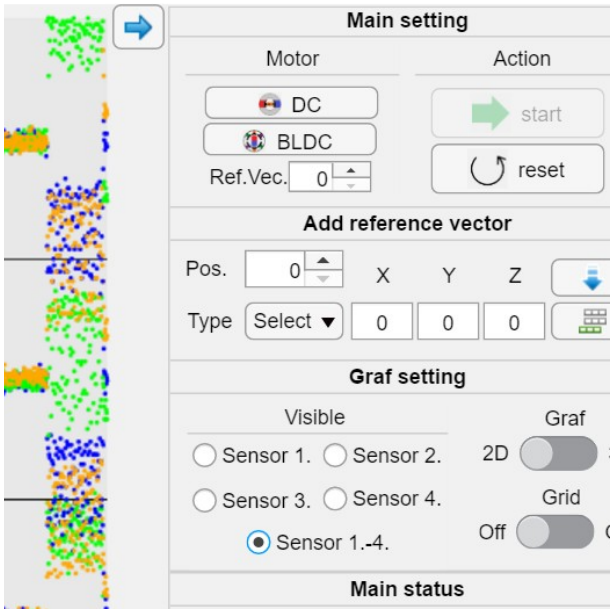






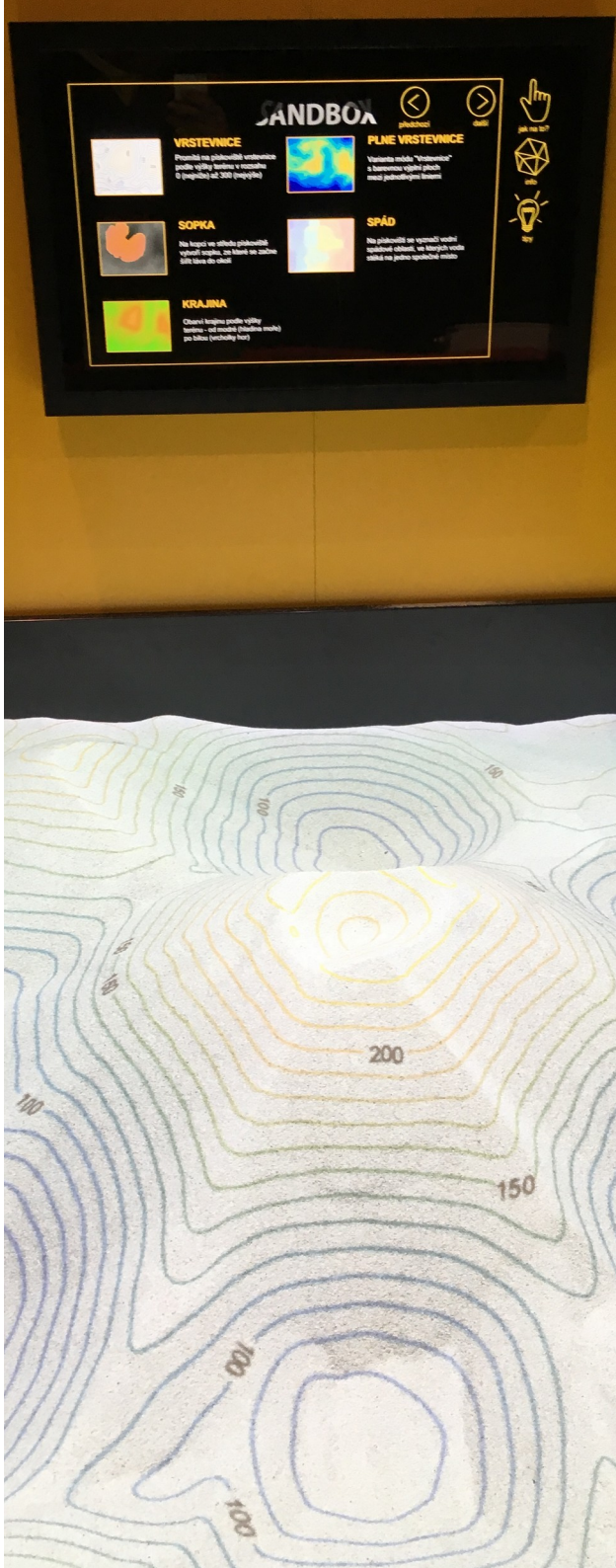
FRAMEWORK pro tvorbu komplexních uživatelských aplikací s GUI v MATLABu

R. Grepl, MECHSOFT s.r.o.

Úvod a **MOTIVACE**

- od r. 2016 vytváříme aplikace **v MATLABu**
- stále hledáme cesty jak je psát
  - efektivněji, snadněji, rychleji
  - funkční, uživatelsky a graficky přívětivé
- průběžně vyvíjíme a používáme **NÁSTROJ**, který nám usnadňuje práci.

[MECH]SOFT



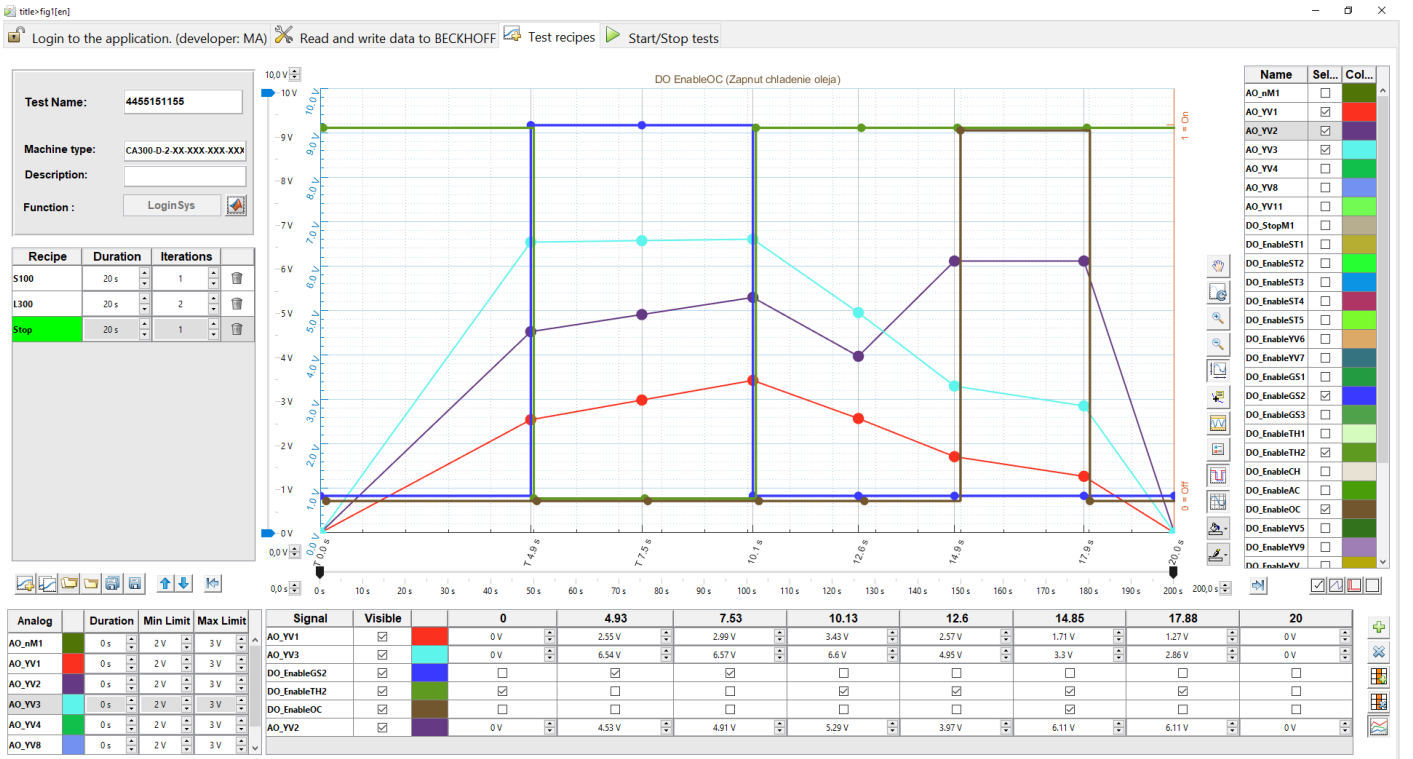
O čem bude řeč… ?

- ukázky aplikací
- současný stav
  - Java-based GUI
  - App Designer
- framework
  - představení, cíle
  - MVC
  - … podrobnosti

[MECH]SOFT

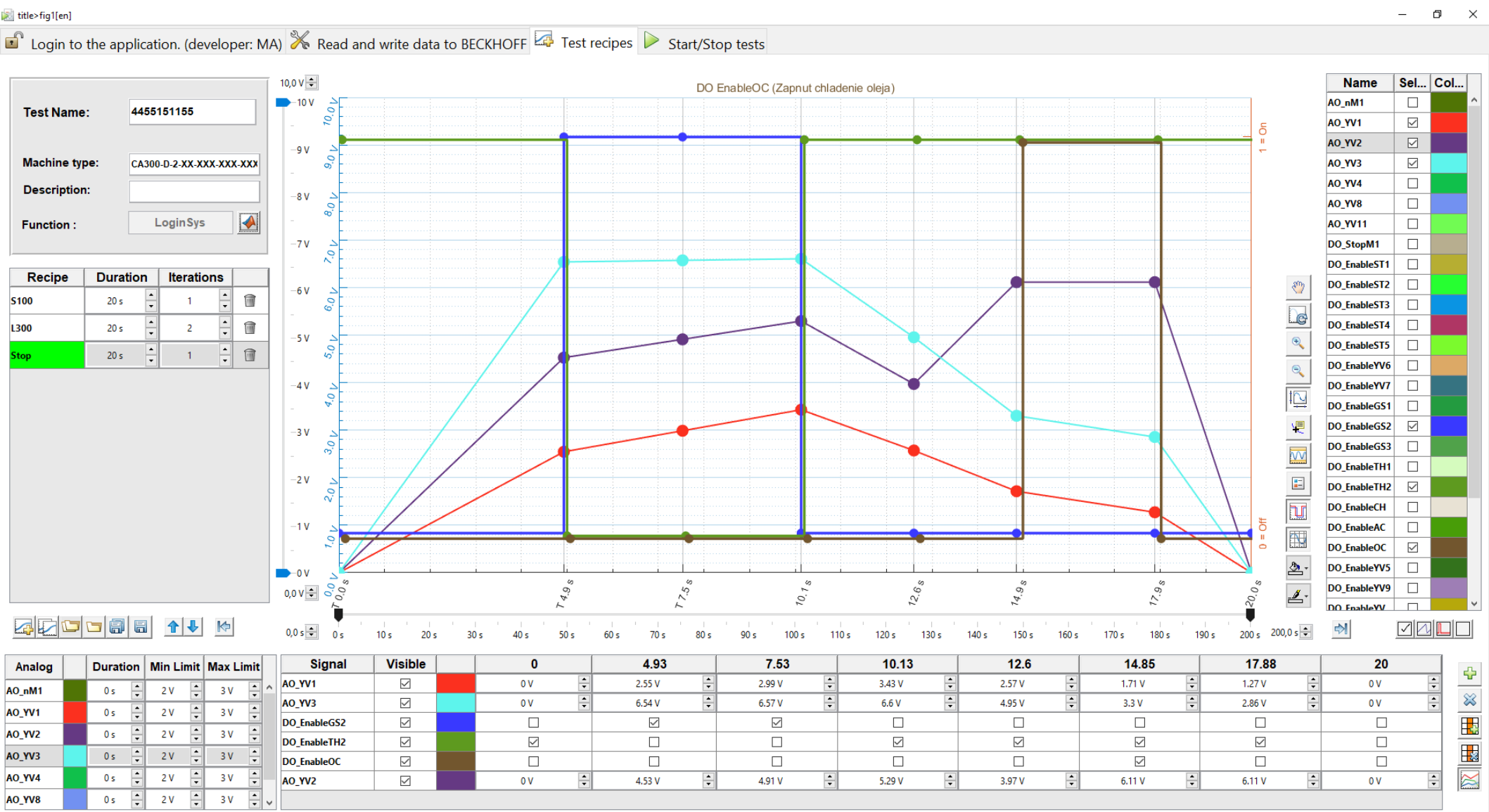
Ukázka aplikace: Hydraulický testovací stand

- HW: BECKHOFF PC/PLC
- ovládácí SW: exe aplikace vytvořená v MATLABu (MATLAB Compiler, MATLAB Runtime)



[MECH]SOFT

Ukázka aplikace: Hydraulický testovací stand

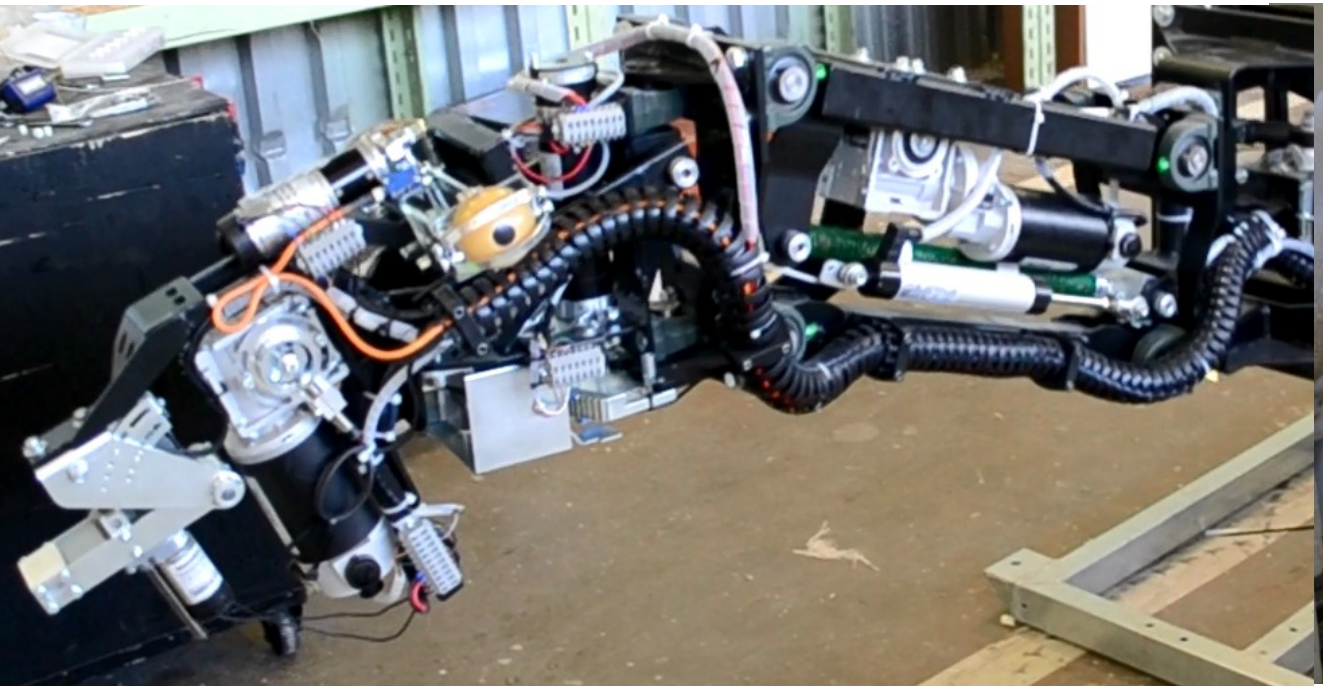


[MECH]SOFT

[video]

Ukázka aplikace: Dino Control

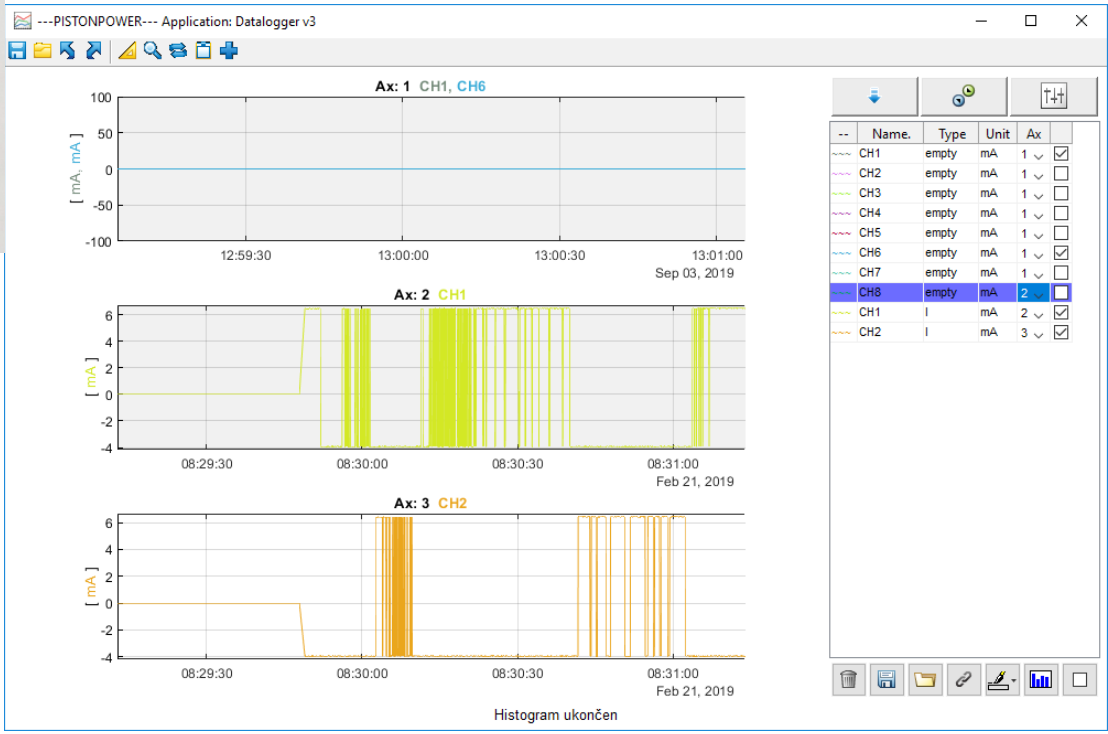
- vlastní HW/ embedded SW
- Desktop SW – „programování“ zvířat



[MECH]SOFT

[video]

Ukázka aplikace: Datalogger



[MECH]SOFT

[video]

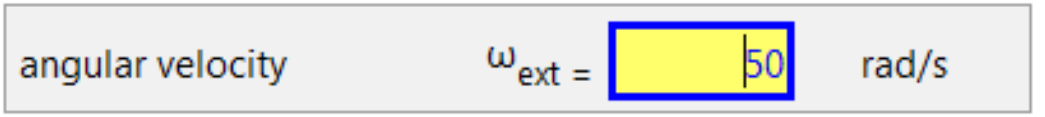
O čem bude řeč… ?

- ukázky aplikací
- současný stav
  - Java-based GUI
  - App Designer
- framework
  - představení, cíle
  - MVC
  - … podrobnosti

[MECH]SOFT

Současný stav

- **Java-based GUI**
  - ~~GUIDE~~
  - uicontrol + Java customization



- **App Designer**
  - uveden 2016b
  - web-based (HTML, localhost webserver, CEF (Chromium Embedded Framework), Dojo Javascript toolkit)

[MECH]SOFT

Současný stav: Java-based G

- uicontrol + Java customization
- příklad: editační políčko
  - **uicontrol:**
    1 callback
  - **Appdesigner: 2**
    (ValueChangedFcn, ValueChangingFcn)
  - Java: 26 (?)

- **AncestorMovedCallback** – fired when one of the compon
- **AncestorAddedCallback** – fired when one of the compone
- **AncestorRemovedCallback** – fired when one of the comp
- **AncestorResizedCallback** – fired when one of the compor
- **ComponentAddedCallback** – fired when a sub-componen
- **ComponentHiddenCallback** – fired when the component
- **ComponentMovedCallback** – fired when the component i
  never fire for them: it does not fire when the container mov
- **ComponentRemovedCallback** – fired when a sub-compor
- **ComponentResizedCallback** – fired when the component
- **ComponentShownCallback** – fired when the component i
- **FocusGainedCallback** – fired when the component gains
- **FocusLostCallback** – fired when the component loses focu
- **HierarchyChangedCallback** – fired when the component
- **KeyPressedCallback** – fired continuously when any keybo
  the specific key and modifiers (Alt, Shift, Ctrl, …) that wer
- **KeyReleasedCallback** – fired when a keyboard button was
  Shift, Ctrl, …) that were pressed. Compare: KeyPressedCal
- **KeyTypedCallback** – similar to KeyPressedCallback, but
  fire twice (Shift, 'A') but KeyTypedCallback will only fire
- **MouseClickedCallback** – fired when a mouse button is pr
  component's bounds, the event will not fire. The figure's 'S
  MousePressedCallback, MouseReleasedCallback.
- **MouseDraggedCallback** – fired continuously when the mo
  beyond the component's bounds. The callback event's meta
  Compare: MouseMovedCallback
- **MouseEnteredCallback** – fired when the mouse is moved
- **MouseExitedCallback** – fired when the mouse is moved (
- **MouseMovedCallback** – fired continuously when the mou
  delta-y of the movement (positive for x-right/y-down; nega
- **MousePressedCallback** – fired immediately when the mou
  contain the click location within the component's bounds.
- **MouseReleasedCallback** – fired immediately when the mo
  within the component's bounds. Compare: MousePressedC
- **MouseWheelMovedCallback** – fired immediately when th
- **PropertyChangeCallback** – fired when one of the compor
  modifying the component's callback properties.
- **VetoableChangeCallback** – fired upon a constrained prop
  Swing components, only JInternalFrame actually declares

[MECH]SOFT

Současný stav: App Designer

- **App Designer** (* 2016b)
  - drag&drop, použitelné pouze pro velmi malé aplikace
  - OOP-based, ale nelze jednoduše oddělit M/V (přímo v AD nelze používat Eventy, nelze editovat část kódu)
  - **lze ale vykopírovat automaticky vygenerovaný kód a pracovat s příkazy** (podobně jako dříve s uicontrol)
  - je **naděje**, že TMW umožní přístup do HTML/JS

    (článek Loren Shure „The State of App Building in MATLAB“, 07/2018)

[MECH]SOFT

Současný stav: Shrnutí

- **MATLAB + Java**
  - téměř neomezené možnosti pro funkcionalitu aplikace, rozumný vzhled
  - od verze 2019b (asi) nelze použít (javaframe)
- **App Designer**
  - moderní (?) technologie
  - zatím stále hodně omezení

O čem bude řeč… ?

- ukázky aplikací
- současný stav
  - Java-based GUI
  - App Designer
- framework
  - představení, cíle
  - MVC
  - … podrobnosti

[MECH]SOFT

„Typický MATLAB user“

- Běžný uživatel MATLABu **umí psát skripty a funkce** a soustředí se na svůj vědecký nebo technický obor – a **není programátorem**.
- Pro vytvoření komplexní aplikace s uživatelským rozhraním (GUI) je však potřeba zvládnout objektové programování (OOP) a v prostředí MATLABu většinou ještě Javu.
- Naší ambicí a cílem je překonat tento rozpor mezi potřebami technika/vědce a potřebou programátorských dovedností a znalostí.

[MECH]SOFT

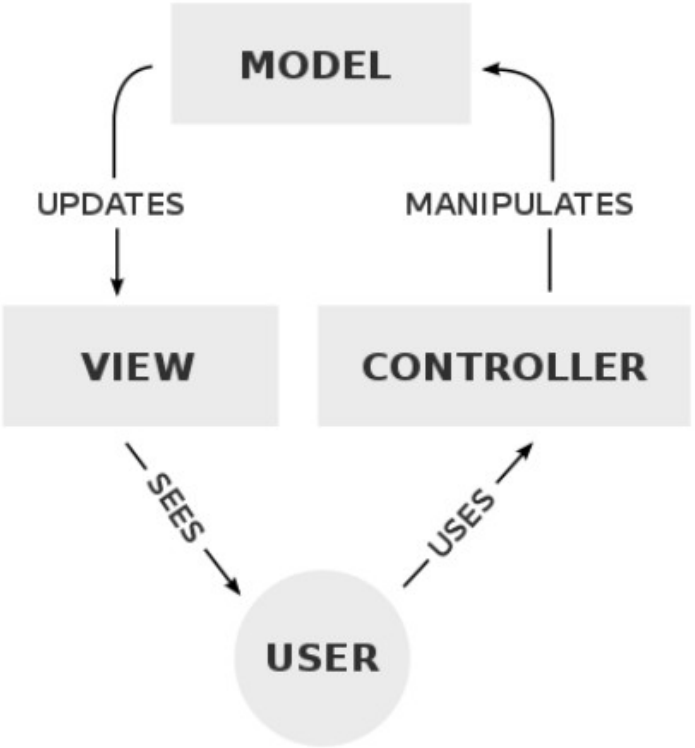
Co je tedy cílem?

- **Běžný uživatel MATLABu bude schopen vytvořit komplexní a přívětivou aplikaci s rozumným úsilím.**
- Jak toho chceme dosáhnout?
  1. MVC – oddělení UI a „algoritmu“
  2. Řešit pozicování prvků, změnu velikosti okna.
  3. Databinding.
  4. Předpřipravené „hezké“ widgety.

[MECH]SOFT

MVC: Oddělení „algoritmu+dat“ (Model) od UI (View)

- Model
  - data, funkcionalita aplikace
  - M musí být schopen fungovat samostatně
- View
  - minimum funkcionality
- Controller
  - zpracovává události z V (i M)



[MECH]SOFT

MVC: View je strom

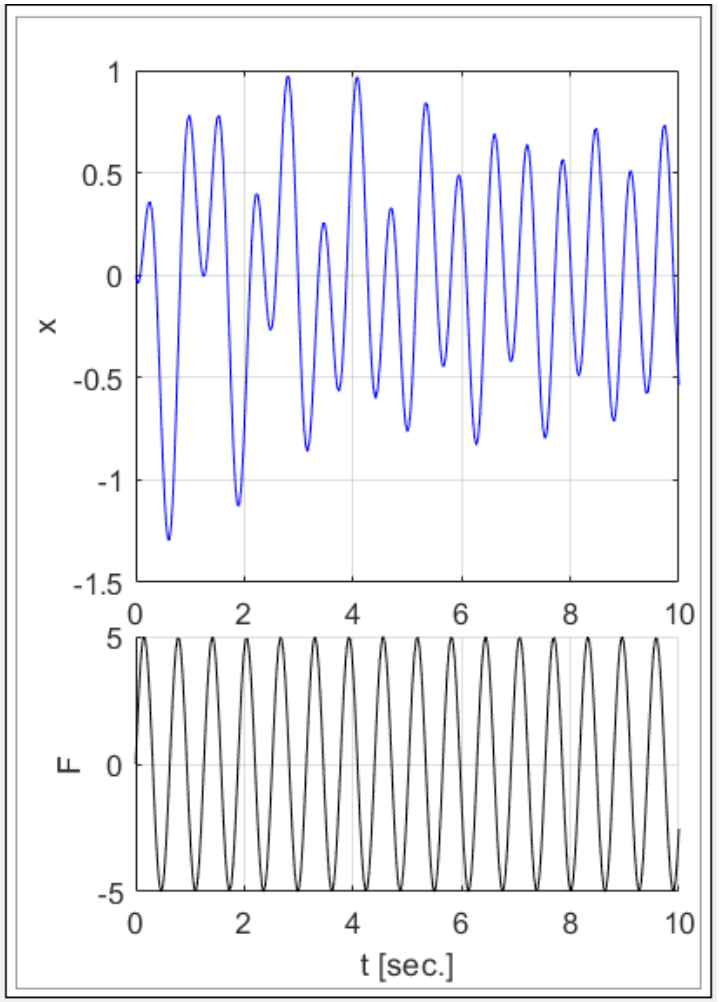
- organizace
- funkce, vlastnosti, rozměry, barvy...

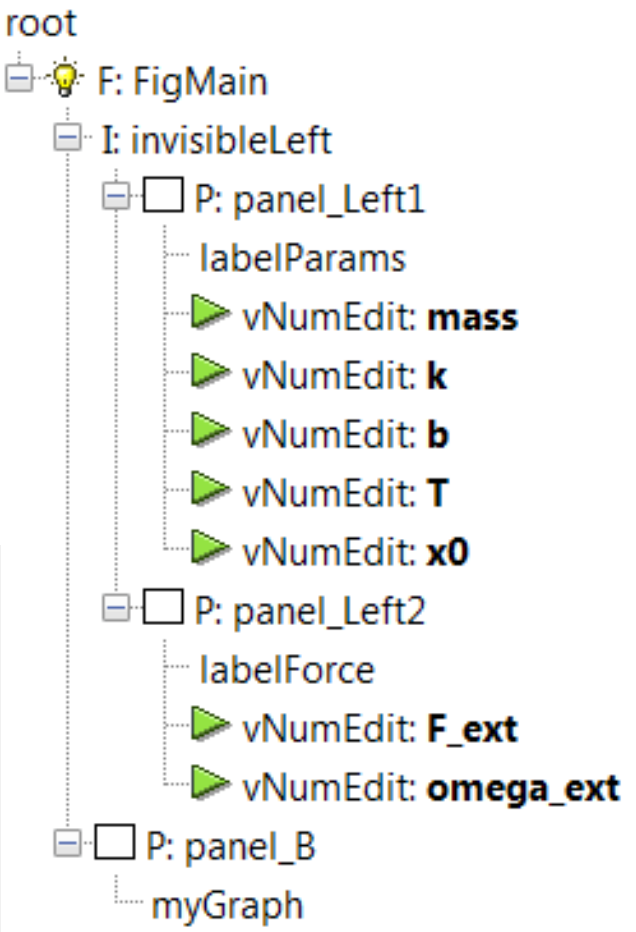
System parameters

mass	m =	5	kg
spring stiffness	k =	100	N/m
viscous damping	b =	2	N*s/m
dry friction	T =	0	N
initial position	x_0 =	0,1	m

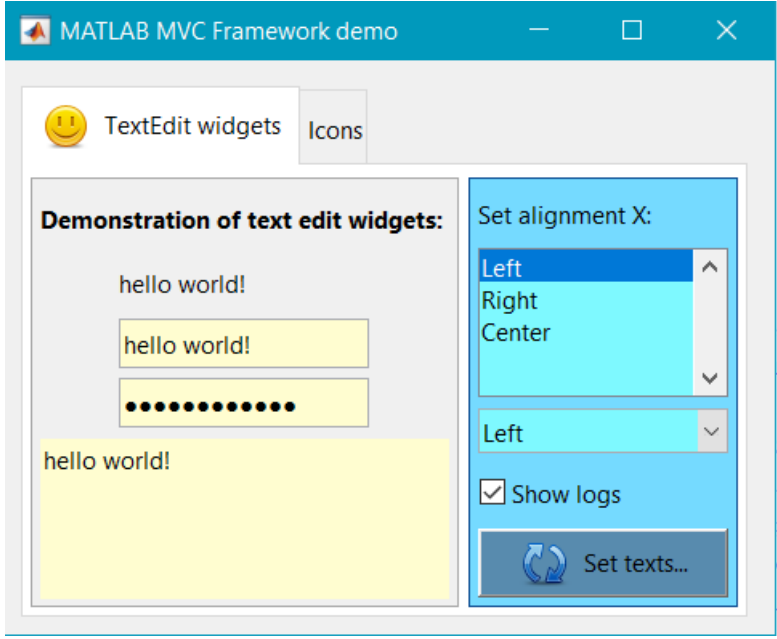
Excitation force parameters

amplitude	F_{ext} =	5	N
angular velocity	ω_{ext} =	10	rad/s





MVC: View – Definice prvků




```
f          = UI.Figure("mainFig");
PanelTabs   = UI.Tabs(f, "PanelTabs");
PanelLeft   = UI.Panel_LR(PanelTabs,"PanelLeft", …
   "txt","TextEdit widgets", "iconFileName","smiley.png");
L1         = UI.Label(PanelLeft,"L1", …
    "txt","Demonstration of text edit widgets:");
L2         = UI.Label(PanelLeft, "T1");
T1a        = UI.TextEdit(PanelLeft, "T1");
T1b        = UI.Password(PanelLeft, "T1");
T1c        = UI.TextEditArea(PanelLeft, "T1");
```

[MECH]SOFT

ViewNode – uzel stromu

- View = strom uzlů.
- struktura uzlu (objekt **ViewNode**) je tato:
  - **box** - rozměry, flexibilita, pozicování
  - **uielement** - render (MATLAB, Java, AppDesigner)
  - **data** - properties, vazba na Model

- Framework je tedy „univerzální“ pokud jde o použitou technologii (Java, App Designer).

[MECH]SOFT

MVC: Co na straně Modelu?

- v Modelu definujeme datové objekty (Piny)
- příklad:
  `myModel.addPin("T1",DATAOBJNAME.string_scalar);`
- Tyto objekty (Piny) jsou propojeny s View pomocí klíče (`"T1"`).
- Piny zajišťují mimo jiné validaci dat.
- `myModel.pins.T1.txt = "ahoj světe";`

property

struktura zajišťující snadný přístup k objektu (Pinu) (používá Key)

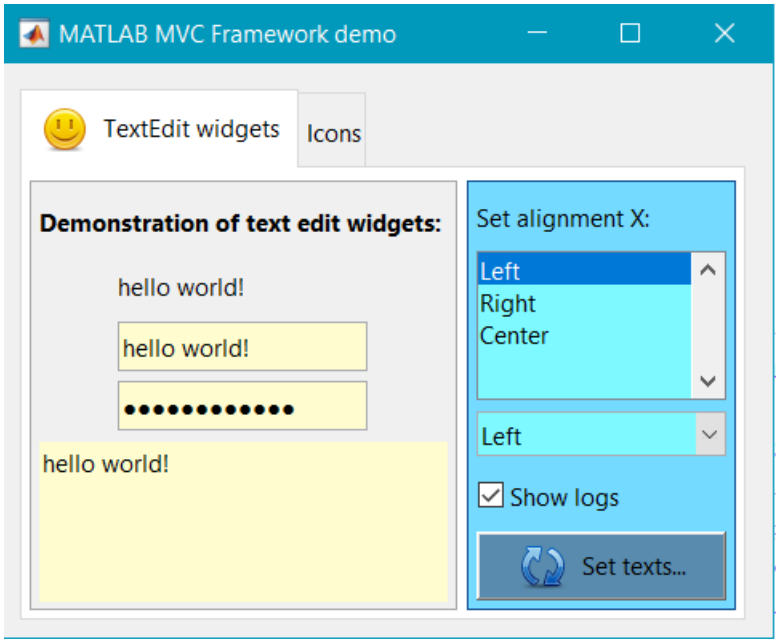
[MECH]SOFT

MVC: ukázka aplikace

- Zápis dat do M způsobí změnu V:

```
m.pins.T1.txt = "Jaké dneska bude pivo?"
```

- akce ve V sdělena C a ten provede reakci



[MECH]SOFT

Framework – Přehled klíčových vlastností

1| MVC

2| automatické pozicování prvků (flow), responzivní

3| Data Binding = automatická vazba mezi daty v M a V

4| uživatel může definovat data M i V v Excelu

5| Undo/Redo funkcionalita

6| rozměry, barvy, fonty … - inspirace CSS (strom)

7| podpora technických výpočtů, fyzikální jednotky

8| ukládání dat M

9| podpora reportování do HTML (PDF)

10| možnost pracovat jak s AppDesignerem tak s M+Java.

[MECH]SOFT

Závěrem...

- MATLAB je skvělým nástrojem pro řešení technických/inženýrských (a jistě i jiných) úloh.
- Občas je potřebné a užitečné vyrobit aplikaci s GUI.
- Snažíme se najít cestu, jak toto maximálně zjednodušit a umožnit „běžnému uživateli“ vytvořit i komplexní dobře udržovatelnou aplikaci.

[MECH]SOFT

Otázky? Zpětná vazba?

1| Nyní?

2| Později tady na TCC?

3| Email, telefon…

Robert Grepl

MECHSOFT s.r.o.
info@mechsoft.cz
M: 732 542 500
U Vodárny 2, 616 00, Brno

[MECH]SOFT