

# EFEKTIVNÍ NÁVRH FUZZY SYSTÉMŮ V PROSTŘEDÍ FUZZYDESIGNER, MATLAB A SIMULINK

Renata Pytelková, Jan Kolínský, Petr Horáček

ProTyS a.s.

**Abstrakt:** *FuzzyDesigner* je nový programový balík určený pro návrh a implementaci fuzzy systémů. Přináší nový přístup, kterým se odlišuje od jiných programových prostředí určených pro návrh fuzzy systémů. Tento přístup je založen na blokově orientované struktuře fuzzy systému, díky které je možné sestavit libovolný fuzzy systém šitý přímo na míru dané úlohy. *FuzzyDesigner* je otevřený aplikační program propojitelný s dalšími řídicími, vizualizačními a informačními systémy nebo průmyslovými programovatelnými automaty. Umožňuje uživateli vygenerovat navržený fuzzy systém jako kód v různých programovacích jazycích a použít jej ve své aplikaci. *FuzzyDesigner* generuje ANSI C, C#, Java, a několik typů funkcí pro programové prostředí Matlab a Simulink. Příspěvek ukazuje různé možnosti propojení fuzzy systému navrženého ve FuzzyDesigneru s programovým prostředím Matlab a Simulink. Jedná se o přímé vygenerování fuzzy systému pro jeho další využití v Matlabu nebo Simulinku, a to jak v programovém jazyce Matlabu, tak jako C-MEX funkce. K vlastnímu výpočtu (vyhodnocení) fuzzy systému v tomto případě dochází v Matlabu. Další možností je vygenerování S-funkcí, zabezpečujících DDE nebo OPC komunikaci Matlabu s FuzzyDesignerem. V tomto případě je fuzzy systém vyhodnocen ve FuzzyDesigneru a mezi Matlabem a FuzzyDesignerem dochází pouze k výměně vstupních a výstupních dat. Poslední možností je využití tzv. funkčních komponent typu dynamicky linkovaných knihoven DLL nebo COM poskytovaných FuzzyDesignerem.

## Úvod

Při návrhu řídicího systému se setkáváme se situacemi, ve kterých lze s výhodou využít řešení ve formě pravidel: *Jestliže ... potom ...*, tedy fuzzy systémy, které umožňují převod vágně popsané znalosti v jazykové formě pomocí implikačních pravidel na matematická zobrazení. Typickým příkladem takových znalostí je jazykový popis závislosti mezi veličinami určitého systému, popis řídicí strategie či klasifikace objektů a situací do tříd, supervize stávajícího řídicího systému nebo beznárazové přepínání mezi několika řídicími strategiemi.



Návrh a realizaci těchto systémů umožňuje nový programový balík *FuzzyDesigner verze 1.01*, vyvinutý firmou ProTyS, a.s.

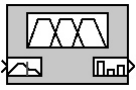


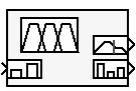

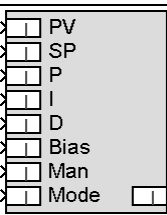
Základní otázkou, kterou si jistě položí každý uživatel, je zda a jakým způsobem je možné navržený fuzzy systém integrovat do již existujícího řídicího systému nebo libovolné aplikace. Těmito aplikacemi mohou být typicky programy typu MS Excel, Matlab, Visual Basic, průmyslové řídicí a vizualizační systémy typu SCADA/HMI, systémy s funkcemi softPLC apod.

Příspěvek stručně seznamuje s vlastním prostředím aplikace *FuzzyDesigner* a dále ukazuje možnosti jeho integrace do programového prostředí Matlab a Simulink.

## FuzzyDesigner

*FuzzyDesigner* je softwarový produkt, určený pro návrh fuzzy systémů. Obsahuje knihovnu komponent, které uživateli umožňují návrh komplexních hierarchických systémů. Knihovna obsahuje následující komponenty:

	<b>Vstupní port</b>	definuje vstupy fuzzy systému, umožňuje předem zpracovat signál s použitím filtru
	<b>Výstupní port</b>	definuje výstupy fuzzy systému

	<b>Vstupní jazyková proměnná</b>	slouží k návrhu funkcí příslušnosti pro vstupní proměnné fuzzy systému, probíhá zde fuzzifikace vstupního signálu proměnné
	<b>Pravidlový blok</b>	slouží k návrhu pravidel a počítá aktivace fuzzy termů výstupních proměnných ze závěrů pravidel na základě vstupních proměnných z předpokladů pravidel
	<b>Vnitřní jazyková proměnná</b>	je určena pro spojování pravidlových bloků v hierarchických systémech
	<b>Výstupní jazyková proměnná</b>	je určena k návrhu fuzzy množin výstupních proměnných, do komponenty vstupují aktivace jednotlivých fuzzy množin z jednotlivých pravidlových bloků, pomocí definované defuzzifikace se vypočítají ostré výstupní hodnoty
	<b>Výstupní Takagi-Sugeno proměnná</b>	je určena pro návrh Takagi-Sugeno fuzzy systémů. Obsahuje parametry lineárních funkcí v závěrech pravidel Takagi-Sugeno systémů a na základě vstupních aktivací jednotlivých funkcí vypočítává hodnotu výstupu komponenty.
	<b>PID regulátor</b>	průmyslové verze PID regulátoru.

Možnost „rozložení“ celého fuzzy systému do jednotlivých komponent dovoluje uživateli sestavit si libovolný fuzzy systém přesně podle svých potřeb. Díky této stavebnicové koncepci je možné vytvářet fuzzy systémy s hierarchickou vnitřní strukturou pravidel. Složitý fuzzy systém je tak rozložen na menší a jednodušší části, které se snáze navrhují a ladí. Snižuje se celkový počet pravidel a zvyšuje přehlednost fuzzy systému jako celku.

*FuzzyDesigner* umožňuje čtení i zápis jak ostrých vstupních a výstupních hodnot, tak i fuzzy množin ve formě seznamu zlomových bodů po částech lineárních funkcí příslušnosti. Funkce příslušnosti definované pro jednotlivé jazykové proměnné mohou být následujícího typu: lichoběžník, inverzní lichoběžník, S-křivka, inverzní S-křivka a singleton.

*FuzzyDesigner* podporuje dvě inferenční metody – Mamdaniho inference a inference s fuzzy aritmetikou, dvě t-normy – minimum a součin a 5 defuzzifikačních metod – dvě varianty metody těžiště a tři vycházející z maxima funkcí příslušnosti výstupních fuzzy množin.

Výsledný fuzzy je možné downloadovat do průmyslových automatů řady Logix5000, SLC 500, PLC-5 a SoftLogix5 od firmy Rockwell Automation, propojit pomocí průmyslového OPC nebo DDE protokolu s jinými aplikacemi, vygenerovat fuzzy systém v jazyce C, C#, Java nebo kód pro Matlab. *FuzzyDesigner* též poskytuje uživateli samostatné komponenty typu DLL/COM a .NET assembly, které může použít ve svých aplikacích.

*FuzzyDesigner* nabízí také několik možností analýzy navrženého fuzzy systému. Jednou z možností je sledování okamžitých hodnot všech parametrů (hodnoty vstupů a výstupů, aktivace jednotlivých funkcí příslušnosti, aktivace pravidel, apod.) celého systému již během jeho návrhu. To uživateli umožňuje snadnější návrh i okamžitou kontrolu chování nejen celého fuzzy systému, ale také jeho jednotlivých částí. Závislost jednotlivých proměnných je možné graficky znázornit pomocí 2D a 3D grafu. Grafy obsahují širokou paletu volitelných funkcí a parametrů. Dále je možné přepnout do tzv. simulačního režimu, ve kterém může uživatel sledovat vliv změny jednotlivých proměnných na ostatní. „Běžící“ fuzzy systém ve *FuzzyDesigneru* je možné též on-line monitorovat a ukládat do souboru hodnoty uživatelem vybraných proměnných.

## Fuzzy systém jako kód pro Matlab

Jedna z možností, jak fuzzy systém navržený ve FuzzyDesigneru integrovat do programového prostředí Matlab, je jeho vygenerování přímo jako kód v jazyce Matlab (syntaxe odpovídá verzi Matlab 5.x). V tomto případě *FuzzyDesigner* vygeneruje celkem 3 různé soubory:

- evalfuz.m* – soubor, který je nezávislý na aktuálním generovaném fuzzy systému, slouží k vlastnímu výpočtu (vyhodnocení) fuzzy systému
- projekt.m* – soubor obsahující parametry a informace o struktuře fuzzy systému, slouží k vytvoření fuzzy systému v Matlabu
- projekt\_s.m* – soubor odpovídající aktuálnímu fuzzy systému pro použití v Simulinku

Funkce *evalfuz.m* obsahuje následující parametry:

```
function [vystup, novystav] = evalfuz(fuzzysystem, vstup, stav)
```

Význam jednotlivých parametrů je následující:

- fuzzysystem* – vlastní fuzzy systém
- vstup* – vektor vstupních hodnot, počet prvků musí být shodný s počtem vstupních portů fuzzy systému
- stav* – vektor vnitřních stavů, parametr se používá pouze v případě dynamických fuzzy systémů
- vystup* – vektor výstupních hodnot, počet prvků musí být shodný s počtem výstupních portů fuzzy systému
- novystav* – vektor “nových” vnitřních stavů po vyhodnocení fuzzy systému

Příklad volání statického fuzzy systému:

```
fs = projekt() % vytvoření fuzzy systému jako proměnné  
                typu "structure"  
vstup = [0.1 0.4 0.5] % zadání vstupních hodnot  
vystup = evalfuz(fs, vstup) % vyhodnocení fuzzy systému (výpočet  
                             výstupních hodnot)
```

Příklad volání dynamického fuzzy systému:

```
fs = projekt() % vytvoření fuzzy systému jako proměnné  
                typu "structure"  
vstup = [0.1 0.4 0.5;...;0.2 -0.1 0.4] % zadání vstupních hodnot  
stav = []; % inicializace interních stavů  
for k = 1:N % N je počet řádků matice vstup  
    [vystup, stav] = evalfuz(fs,vstup(k,:),stav) % vyhodnocení fuzzy systému  
end
```

V případě zakomponování fuzzy systému do simulinkového modelu je možné přímo použít funkci *projekt\_s.m*, která v sobě již obsahuje předchozí příkazy.

## Fuzzy systém jako C-MEX funkce

*FuzzyDesigner* umožňuje generovat navržený fuzzy systém také jako C-MEX funkci pro Matlab. V tomto případě *FuzzyDesigner* vygeneruje následující soubory:

- fuzcore.h* – soubor, který je nezávislý na aktuálním generovaném fuzzy systému, slouží k definici konstant a struktur
- fuzcore.c* – soubor, který je nezávislý na aktuálním generovaném fuzzy systému, slouží k vlastnímu výpočtu (vyhodnocení) fuzzy systému
- projekt\_c.c* – soubor obsahující parametry a informace o struktuře fuzzy systému

*projekt\_c.h* – soubor obsahující deklarace funkcí nabízených uživateli  
*projekt.c* – soubor, který obsahuje definici C-MEX funkce  
*projekt\_mexs.m* – s-funkce pro Simulink volající C-MEX funkci

Postup použití takto vygenerovaných souborů v Matlabu je potřeba zkompileovat soubor *projekt.c* do MEX funkce pomocí příkazu v Matlabu *mex*. Tím bude vytvořen soubor *projekt.dll*, který se může dále využít v Matlabu.

Příklad použití C-MEX funkce pro Matlab:

<code>mex projekt.c</code>	<code>% vytvoření souboru projekt.dll</code>
<code>vstup = [0.1 0.4 0.5]</code>	<code>% zadání vstupních hodnot</code>
<code>vystup = projekt(vstup)</code>	<code>% vyhodnocení fuzzy systému</code>

V případě, že se jedná o dynamický fuzzy systém, dochází před prvním zavoláním k inicializaci vnitřních stavů. Pokud je z nějakého důvodu potřeba vnitřní stavy vynulovat, je potřeba MEX funkci odstranit z paměti pomocí příkazu *clear*.

<code>clear projekt</code>	<code>% odstranění MEX funkce z paměti</code>
----------------------------	---

## DDE komunikace mezi FuzzyDesignerem a Matlabem

DDE (Dynamic Data Exchange) protokol je již poměrně starý, ale průmyslovými aplikacemi stále podporovaný.

*FuzzyDesigner* se chová jako DDE server, který umožňuje ostatním aplikacím, v tomto případě Matlabu, připojení a přístup (čtení i zápis) na vstupní a výstupní porty fuzzy systémů. V okamžiku, kdy klientská aplikace zapíše hodnoty na vstupní porty, je daný fuzzy systém automaticky vyhodnocen a vypočítané výstupní hodnoty jsou zpřístupněny pro čtení z výstupních portů.

Pro navázání komunikace klient nejprve specifikuje *jméno aplikace* a tzv. *topic*. V případě FuzzyDesigneru má *jméno aplikace* hodnotu "FuzzyDesigner" a hodnota *topic* je jméno aktuálního fuzzy systému. Po navázání komunikace lze provádět základní operace typu zápis a čtení, při kterých je potřeba specifikovat datový zdroj (pro čtení) a datovou jímku (pro zápis) názvem tzv. DDE *item*. Struktury objektů typu *topic* a *item* definují tzv. jmenný prostor DDE serveru. Každý port fuzzy systému má odpovídající objekt *item* ve jmenném prostoru. Formát zasílaných dat musí být v případě FuzzyDesigneru typu TEXT (ASCII nebo UNICODE). Aplikace *FuzzyDesigner* umožňuje zápis i čtení jak ostrých čísel tak fuzzy množin ve formě seznamu zlomových bodů po částech lineárních fuzzy množin.

*FuzzyDesigner* umožňuje uživateli vygenerování s-funkce pro použití v Simulinku, která zajišťuje DDE komunikaci mezi Matlabem a FuzzyDesignerem. V této funkci je zajištěno navázání komunikace – specifikováno *jméno aplikace* a tzv. *topic*, odeslání aktuálních vstupních hodnot fuzzy systémů do FuzzyDesigneru a dále přečtení výstupních hodnot fuzzy systému z FuzzyDesigneru.

Nutné kroky pro vytvoření DDE komunikace mezi FuzzyDesignerem a Matlabem:

1. Otevření ve FuzzyDesigneru fuzzy systému, který chceme použít v Matlabu.
2. Vygenerování z FuzzyDesigneru (Menu → Tools → Code Generators → MATLAB DDE S-Function) s-funkce pro zajištění DDE komunikace.
3. Vložení této funkce do simulinkového schématu.
4. Spuštění simulace.

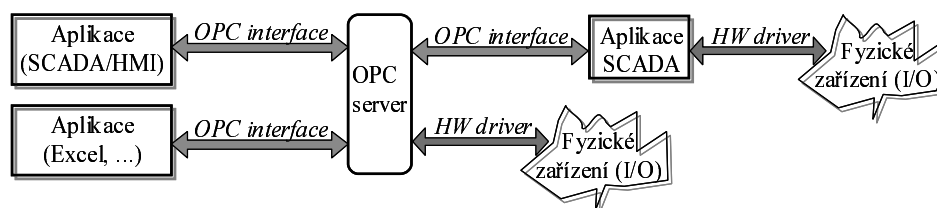
V případě spuštění již dříve vytvořeného simulinkového schématu je vždy před spuštěním simulace potřeba otevřít fuzzy systém ve FuzzyDesigneru.

## OPC komunikace mezi FuzzyDesignerem a Matlabem

OPC (*OLE for Process Control*) je otevřený protokol definovaný skupinou *OPC Foundation* a v současnosti je nejrozšířenějším standardem pro komunikaci průmyslových informačních systémů. Protokol je založen na technologii OLE (Object Linking and Embedding) a COM/DCOM (Distributed Component Object Model) firmy Microsoft. Protokol COM definuje standard pro vývoj objektových

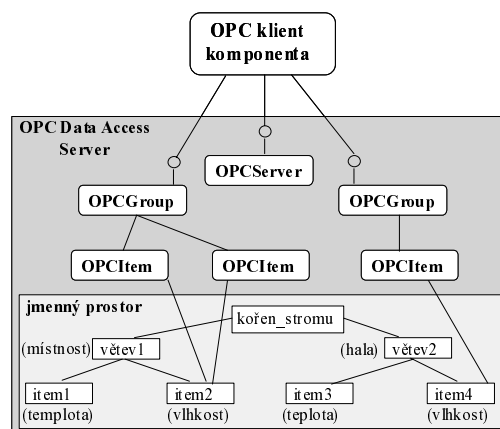
softwarových komponent a jejich interakci na binární úrovni v rámci jednoho procesu. Protokol DCOM, postavený na mechanismu vzdáleného volání procedur RPC (Remote Procedure Call), je pak jeho rozšířením pro komunikace mezi procesy v rámci jednoho počítače i sítě.

Pro výměnu dat reálného času je určen standard *Data Access Custom Interface Standard*. Programová komponenta implementující tento interface se nazývá *OPC DA server*. Tato architektura umožňuje jednotný přístup k aplikačním a procesním datům systémů různých výrobců (Obr. 1). Klientské programy mohou provádět synchronní operace čtení dat, kdy server okamžitě poskytuje aktuální data podle požadavku z bufferu nebo zařízení. Při synchronním zápisu jsou přenesená data okamžitě aplikována. Klient má možnost být pomocí událostí informován o změnách hodnot vybraných dat a pro tento případ může uvést pásmo necitlivost v %. Asynchronní operací čtení je možno zaslat požadavek k získání aktuálních dat, jakmile jsou data k dispozici jsou zpětně zaslána klientovi formou událostí. Ten tak není blokován po dobu vyřizování požadavku. Podobně při operaci asynchronního zápisu není klient blokován v průběhu aplikování zapsaných dat, ale je zpětně informován o úspěšném či neúspěšném provedení akce.



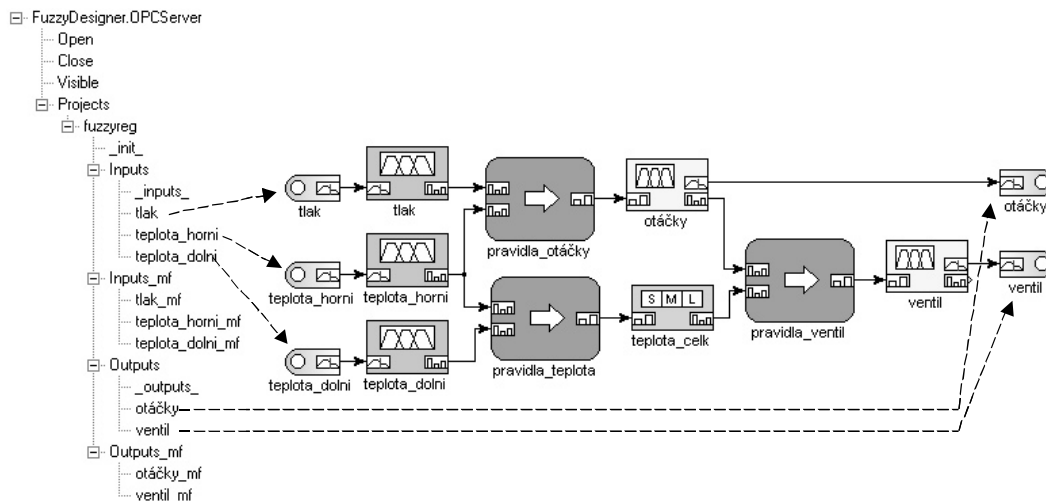
Obr. 1 OPC klient/server interface.

Data poskytovaná serverem jsou k dispozici v hierarchické stromové struktuře nazývané jmenný (adresní) prostor serveru (Obr. 2). Klienti mají možnost tuto strukturu procházet. Listy tohoto stromu jsou tzv. *OPCItem* objekty poskytující základní funkcionalitu pro čtení a zápis dat. OPC klient po připojení k serveru vytvoří objekt *OPCGroup*, do kterého poté vkládá *OPCItem* objekty specifikované jednoznačným jménem (*ItemID*) definovaným ve jmenném prostoru serveru. Přenos dat mezi objekty se uskutečňuje na základě strukturovaného datového typu *VARIANT* (string, int, float, ...), jehož konkrétní hodnota může být klientem volitelně měněna.



Obr. 2 Vztah jmenného prostoru a hierarchie objektů OPC serveru.

Součástí aplikace *FuzzyDesigner* je OPC server poskytující operace pro otevírání fuzzy systémů z disku, čtení a zápis hodnot portů fuzzy systému a jeho automatické vyhodnocení po operaci zápisu. Jmenný prostor serveru poskytuje *OPCItem* objekty pro jednotlivé porty aktuálně otevřených fuzzy systémů (Obr. 3). Pro každý port je též k dispozici *OPCItem* pro čtení a zápis po částech lineárních fuzzy množin.



Obr. 3 Jmenný prostor aplikace *FuzzyDesigner*.

Stejně jako v případě DDE komunikace *FuzzyDesigner* umožňuje uživateli také vygenerování s-funkce pro použití v Simulinku, která zajišťuje OPC komunikaci mezi Matlabem a *FuzzyDesignerem*. V tomto případě opět vlastní vyhodnocení fuzzy systémů probíhá ve *FuzzyDesigneru* na základě vstupních dat získaných z Matlabu.

V této funkci je zajištěno navázání komunikace – připojení k *FuzzyDesigneru* OPC serveru a otevření projektu, odeslání aktuálních vstupních hodnot fuzzy systémů do *FuzzyDesigneru* a dále přečtení výstupních hodnot fuzzy systému z *FuzzyDesigneru*.

Základní výhodou OPC protokolu je jeho stabilita, rychlost a bezpečnost v meziprocením komunikaci v rámci jednoho počítače i lokální sítě.

Nutné kroky pro vytvoření OPC komunikace mezi *FuzzyDesignerem* a Matlabem:

1. Otevření ve *FuzzyDesigneru* fuzzy systému, který chceme použít v Matlabu.
2. Vygenerování z *FuzzyDesigneru* (Menu → Tools → Code Generators → MATLAB OPC S-Function) s-funkce pro zajištění OPC komunikace.
3. Vložení této funkce do simulinkového schématu.
4. Spuštění simulace.

V případě OPC komunikace je možné nastavit při spuštění simulace již dříve vytvořeného simulinkového schématu automatické otevření fuzzy systému ve *FuzzyDesigneru*. Příslušná část kódu je v generované s-funkci již připravena jako komentář. Pro automatické otevření je nutno vymazat úvodní symbol komentáře '%', popřípadě nastavit správnou cestu k souboru s daným fuzzy systémem. Viz. následující výpis.

```

%===== enables opening and closing projects: =====
%global OPC_close_ball
%global OPC_ball_path
%OPC_ball_path = [cd '\ball.fsp']; % specify project path

```

### Fuzzy modul jako samostatná komponenta

Řada vývojových prostředí umožňuje integraci funkčních komponent typu dynamicky linkovaných knihoven DLL nebo COM(ActiveX). Tyto komponenty jsou pak mapovány do adresního prostoru volajícího procesu.

*FuzzyDesigner* poskytuje uživateli komponenty, které mu umožňují nahrání a vyhodnocení fuzzy systémů aniž by musel instalovat a pouštět vlastní aplikaci *FuzzyDesigner*.

#### Samostatná komponenta .NET assembly a COM

V podadresáři NET aplikace *FuzzyDesigner* se nachází .NET assembly „FuzzyDesignerNET.dll“, která je také zaregistrována jako COM komponenta

„FuzzyDesignerNET.FuzzySystem“, dále „FDesignerOPC\_NETMonitor.dll“, která umožňuje on-line monitorování běžícího fuzzy systému a dále „Interop.FDesignerOPC\_NETMonitorLib.dll“ která má zaručuje propojení mezi .NET a COM serverem. Jestliže chce uživatel tyto komponenty použít na počítači, kde není nainstalován *FuzzyDesigner*, stačí tyto komponenty jednoduše zkopírovat na tento počítač a zaregistrovat je pomocí souboru „REGMonitorServer.bat“. Je nutné, aby na tomto počítači byl instalován Microsoft.NET Framework. Komponenta „FuzzyDesignerNET.dll“ může být též volitelně registrována jako komponenta typu COM (typová knihovna „FuzzyDesignerNET.tlb“).

Komponenta poskytuje uživateli následující funkce: nahrání a vymazání fuzzy systému do/z paměti, získání počtu vstupních a výstupních portů fuzzy systému, zápis hodnot („ostrých“ čísel i fuzzy množin) do vstupních portů, čtení hodnot z výstupních portů, nulování vnitřních stavů a vyhodnocení fuzzy systému.

Ukázky volání jednotlivých funkcí je v následujícím přehledu.

<code>FuzzyDesignerCOM_Server = actxserver('FuzzyDesignerNET.FuzzySystem')</code>	vytvoření COM objektu propojeného s .NET komponentou
<code>invoke(FuzzyDesignerCOM_Server, 'StartMonitoringServer')</code>	spuštění OPC serveru pro monitorování
<code>invoke(FuzzyDesignerCOM_Server, 'LoadSystem', [cd '\coupleddrives.fsp'])</code>	načtení fuzzy systému do paměti
<code>m = invoke(FuzzyDesignerCOM_Server, 'InputsCount')</code>	získání počtu vstupních portů
<code>n = invoke(FuzzyDesignerCOM_Server, 'OutputsCount')</code>	získání počtu výstupních portů
<code>for i=1:length(u)     res = invoke(FuzzyDesignerCOM_Server, 'SetInput', i-1, u(i)); end</code>	zápis hodnot do vstupních portů
<code>invoke(FuzzyDesignerCOM_Server, 'EvaluateSystem')</code>	vyhodnocení fuzzy systému
<code>y = invoke(FuzzyDesignerCOM_Server, 'GetOutput', 0)</code>	přečtení hodnot z výstupních portů
<code>invoke(FuzzyDesignerCOM_Server, 'ResetState')</code>	vynulování vnitřních stavů
<code>invoke(FuzzyDesignerCOM_Server, 'UnloadSystem')</code>	vymazání fuzzy systému z paměti
<code>invoke(FuzzyDesignerCOM_Server, 'TerminateMonitoringServer')</code>	ukončení OPC serveru pro monitorování
<code>release(FuzzyDesignerCOM_Server)</code>	uvolnění paměti

### Samostatná komponenta DLL a COM

*FuzzyDesigner* poskytuje také samostatné komponenty nezávislé na Microsoft.NET Framework. V podadresáři DLL\_COM aplikace *FuzzyDesigner* se nachází COM komponenta „FuzzyDesignerCOM.dll“, která je zaregistrována jako „FuzzyDesignerCOM.CFuzzySystem“. Komponenta obsahuje interní OPC server, který umožňuje on-line monitorování a ladění z prostředí *FuzzyDesigner*. Jestliže chce uživatel tyto komponenty použít na počítači, kde není nainstalován *FuzzyDesigner*, opět stačí tyto komponenty jednoduše zkopírovat na tento počítač a zaregistrovat je pomocí souboru „REGFuzzyDesignerCOM.bat“.

Volání jednotlivých funkcí je stejné, jako v předchozím případě, rozdíl je ve vytvoření COM objektu:

<code>FuzzyDesignerCOM_Server = actxserver('FuzzyDesignerCOM.FuzzySystem')</code>	vytvoření COM objektu
---	-----------------------

### Závěr

Příspěvek ukázal možnosti integrace fuzzy systémů navržených v aplikaci *FuzzyDesigner* do programového prostředí Matlab a Simulink. *FuzzyDesigner* umožňuje uživateli přímé vygenerování navrženého fuzzy systému v kódu pro Matlab (m-file, s-funkce, C-MEX funkce). V tomto případě dochází k vlastnímu vyhodnocení fuzzy systému přímo v Matlabu a není závislé na aplikaci

*FuzzyDesigner*. Druhá možnost je založena na komunikaci mezi FuzzyDesignerem a Matlabem pomocí DDE nebo OPC serveru, popřípadě mezi funkční komponentou poskytovanou FuzzyDesignerem a Matlabem. V tomto případě dochází k vyhodnocování fuzzy systému ve FuzzyDesigneru (popřípadě funkční komponentě). Tyto komunikační možnosti dovolují propojit aplikaci *FuzzyDesigner* nejenom s programovým prostředím Matlab, ale i s dalšími řídicími, vizualizačními a informačními systémy nebo průmyslovými programovatelnými automaty.

**Kontakt:**

ProTyS, a.s., Na kopečku 1, 180 00 Praha 8

Tel: 224 917 207-8, Fax: 224 917 209

www: <http://www.protys.cz>, e-mail: [FuzzyDesigner@protys.cz](mailto:FuzzyDesigner@protys.cz)