

SPECIFIC UTILIZATION OF MICROSOFT VISUAL BASIC FOR APPLICATION WITH PRINCIPLES OF SYSTEM MODELING

Tomáš BAROT

Abstract: The article is focused on utilization of programming language Microsoft Visual Basic for Application for educational purposes. These software possibilities are not mentioned as innovation, but it is pointed out on their advantages for area of student education. It is introduced utilization of program tools of simulation and modeling of physical effects described as discrete systems. Mathematical calculations are presented in the example which can be used and implemented by teachers. Inclusion of mathematical computation in the Excel formula allows to examine modeled systems from the view of behavior. The support of education can be used as an inspiration of mathematical and physical solution by informatics.

Key words: Microsoft Visual Basic for Application, modeling, simulation, education, applied informatic, discrete systems.

SPECIFICKÉ VYUŽITÍ MICROSOFT VISUAL BASIC FOR APPLICATION S PRINCIPY MODELOVÁNÍ SYSTÉMŮ

Resumé: Článek je zaměřen na využití programovacího jazyka Microsoft Visual Basic for Application pro výukové účely. Tyto softwarové možnosti nejsou uváděny jako novinka, ale je poukázáno na jejich prospěch pro oblast vzdělávání studentů. Je představeno využití programových nástrojů při simulaci a modelování fyzikálních jevů popsaných jako diskrétní systémy. Matematické výpočty jsou uvedeny na příkladu, který mohou vyučující využít a implementovat. Zahrnutí výpočtu do rozhraní vzorce Excelu umožní studentům namodelované systémy zkoumat z pohledu jejich chování. Uvedená podpora výuky může sloužit i jako inspirace matematického a fyzikálního řešení za pomoci informatiky.

Klíčová slova: Microsoft Visual Basic for Application, modelování, simulace, vzdělávání, aplikovaná informatika, diskrétní systémy.

1 Úvod

I když se student s diferenciálními rovnicemi setkává poprvé až v bakalářském stupni vysokoškolského studia, může již na střední škole získat obecné povědomí o tom, jak se jimi dá provést matematický popis reálných systémů. Konkrétně by bylo vhodné přirovnat práci s nimi v analogii s předpisem pro výpočet funkční hodnoty funkce jedné reálné proměnné. V tomto případě je relačně dáno vztahem přiřazení jedné závislé proměnné jedné nezávislé proměnné. Z tohoto přirovnání by mohlo být studentům lépe patrné, že systém funguje na podobné myšlence, kdy je funkci na vstupu systému přiřazena výstupní funkce. Vyjádření této transformace diferenciální rovnicí se však dozví až při univerzitním vzdělání. Pro obecné povědomí jim však může být přínosná informace, že je vstupem celý funkční průběh veličiny a na výstupu systému je vrácen opět časový signál ve formě odezvy. Se vstupními daty lze poté experimentovat na simulačních modelech, např.

v Microsoft Excelu, kdy studenti zkoumají různé odezvy systému fyzikální podstaty. I když netuší jak je transformace vstupů na výstupy konkrétně řešena, uvědomí si, že se nesestává pouze z jedné relace, a to jeden konkrétní vstup - jeden konkrétní výstup, ale libovolný vstup - konkrétní výstup pro daný jedinečný systém. Zobecnění uvedeného principu by bylo samozřejmě více rozměrové modelování systému s matematickým popisem ve tvaru soustavy diferenciálních rovnic.

Z dosavadního pojednání je patrné, že pro tvorbu simulačních modelů popsaných diferenciální rovnicí bude nutný její převod na diskrétní formu, tedy diferenční rovnici, aby bylo možné numerické řešení předpisu výstupního signálu. Systém bude následně pracovat již se vzorky vstupní a výstupní posloupnosti, kde jejich transformačním vztahem bude diskrétní popis.

Pro práci s modelem systému bude potřeba určit vztah pro určení posloupnosti výstupní veličiny. Při odvození vztahu bude uvažován

převod dle pramenů [1], [2]. Pro derivaci by mohla být použita aproximace, v odborných knihách často uváděná jako Eulerova, nýbrž pro velmi malé periody vzorkování mohou nastat numerické problémy ohrožující stabilitu systému a je třeba uvažovat Delta-transformaci a popis systémů Delta-modely [3]. Odvození výstupní rovnice je provedeno ve stavové oblasti, kdy je rovnice vnějšího popisu systému transformována na stavový popis. Jeden systém může mít obecně, z pohledu volby stavových proměnných, jiný stavový popis, ale přesto zcela odpovídající. Stavovým popisem se zabývají zdroje [1], [2], ale především bych doporučoval publikaci [4].

2 a ískání výstupní rovnice diskretní verze systému na základě znalosti jeho spojitého popisu

Předpokládáme matematický model systému ve tvaru lineární diferenciální rovnice. Systém uvažujeme jednorozměrný, tedy s jedním vstupem a jedním výstupem. Podle pramene [5], je proveden převod diferenciální rovnice na stavový popis (1), (2). Použitá symbolika a význam veličin je podobný v rámci zavedeného systému značení napříč zdroji [1] až [7], i když se mohou v drobných ohledech lišit.

$$\mathbf{x}'(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) \quad (1)$$

$$y(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}u(t) \quad (2)$$

Dále je nutné provést převod spojitého stavového popisu (1), (2) na diskretní (3), (4) při simulačním čase (5) s periodou vzorkování T .

$$\mathbf{x}((k+1)T) = e^{\mathbf{A}T}\mathbf{x}(kT) + \int_0^T e^{\mathbf{A}(T-\tau)}\mathbf{B}u(\tau)d\tau \quad (3)$$

$$y(kT) = \mathbf{C}\mathbf{x}(kT) + \mathbf{D}u(kT) \quad (4)$$

$$t = kT, k \in \mathbb{Z}_0^+ \quad (5)$$

Diskretní stavový popis je výhodný i pro mnohorozměrné modely. Stačilo by přeznačit skalární vyjádření veličin na vektory. Transformace soustavy diferenčních rovnic na stavový popis by byla o něco náročnější, ale lze použít algoritmus pro diskretní mnohorozměrné systémy uvedený ve zdroji [2]. Zde bude kladen zřetel pouze na jednorozměrné systémy, kdy stavový popis (3), (4) můžeme přepsat do přehlednější formy (6), (7). Vztahy pro určení nových matic jsou patrné z analogie rovnic (3), (4) a (6), (7). Výstupní matice a matice převodu zůstávají nepozměněny a jsou pouze přeznačeny.

$$\mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{G}u(k) \quad (6)$$

$$y(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{L}u(k) \quad (7)$$

Analytické řešení převodu spojitého stavového popisu na diskretní může být zjednodušen použitím programového prostředí MATLAB. Posloupnost příkazů `spStavovy=ss(A,B,C,D); T=1; diskStavovy=c2d(spStavovy,T)` nám pro zadané spojitě matice vypíše jejich diskretní varianty ve smyslu rovnic (6) a (7).

Závěrečný výpočet výstupu systému je již řešením stavových rovnic, a to pouhým dosazováním posloupností diskretních veličin rekurentním způsobem (8), (9).

Tím jsou stanoveny obecné numerické vztahy, které lze jednoduše implementovat v libovolném programovacím jazyku s rozšířenou množinou algebraických operací pro matice.

$$\mathbf{x}(k) = \mathbf{F}^k \mathbf{x}(0) + \sum_{r=0}^{k-1} \mathbf{F}^{k-r-1} \mathbf{G}u(r) \quad (8)$$

$$y(k) = \mathbf{H}\mathbf{F}^k \mathbf{x}(0) + \sum_{r=0}^{k-1} \mathbf{H}\mathbf{F}^{k-r-1} \mathbf{G}u(r) + \mathbf{L}u(k) \quad (9)$$

3 Příklad stanovení diskretní výstupní rovnice za účelem modelování systému

Jako praktická aplikace je uveden příklad konkrétního určení výstupních rovnic typu (8), (9), lineárního spojitého dynamického systému 2. řádu (10) a (11) s periodou vzorkování 0,25 s. Spojitý stavový popis (15), (16) byl převeden v prostředí MATLAB na diskretní (17), (18) pomocí sady příkazů (Obr.1).

$$y''(t) + 0.25y'(t) + 0.5y(t) = 0.6u(t) \quad (10)$$

$$y'(0) = 0, y(0) = 0 \quad (11)$$

$$x_1(t) = y(t) \quad (12)$$

$$x_2(t) = x_1'(t) = y'(t) \quad (13)$$

$$x_2'(t) = -0.25x_2(t) - 0.5x_1(t) + 0.6u(t) \quad (14)$$

$$\mathbf{x}'(t) = \begin{bmatrix} 0 & 1 \\ -0.5 & -0.25 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 0.6 \end{bmatrix} u(t) \quad (15)$$

$$y(t) = [1 \ 0] \mathbf{x}(t) + [0] u(t) \quad (16)$$

```
A=[0 1; -0.5 -0.25]; B=[0; 0.6]; C=[1 0];
D=[0]; T=0.25; spStavovy=ss(A,B,C,D);
diskrStavovy=c2d(spStavovy,T)
```

Obr 1: Programový kód v prostředí MATLAB pro převod spojitého stavového popisu na diskretní

$$\mathbf{x}(k+1) = \begin{bmatrix} 0.98 & 0.24 \\ -0.12 & 0.93 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.02 \\ 0.15 \end{bmatrix} u(k) \quad (17)$$

$$y(k) = [1 \ 0] \mathbf{x}(k) + [0] u(k) \quad (18)$$

Tvary matic v diskretním stavovém popisu jsou v této podobě patrné a jejich dosazením do rovnic (8), (9) obdržíme konečné rovnice, které implementujeme. Pro určení konkrétního výstupního signálu v okamžicích vzorkování postačuje dosadit do rovnic, v jednotlivých iteracích jejich řešení, hodnoty vstupní posloupnosti a počáteční stavový vektor. Počáteční stav bude v našem příkladu definován např. jako nulový vektor.

4 Možnosti implementace modelování systému

V předkládaném článku je apriorně doporučen Microsoft Excel s integrovaným programovacím jazykem Microsoft Visual Basic for Application. Bližší znalosti byly čerpány ze statí zdrojů [8] až [10]. Pro numerické výpočty lze využít i klasické programovací jazyky. Ve vlastním řešení by muselo být navrženo uživatelské simulační rozhraní a další práce s daty by závisely na vytvořených možnostech. Proto je jednodušší uzavření výpočtů do vzorce a poskytnutí prostoru studentům ve prospěch rozvoje jejich kreativity při modelování. Použitím Excelu se vyhneme komplikovanějším vlastním řešením např. v jazyku C++.

Studenti budou využívat pro práci s modelem rozhraní ve formě vzorce Excelu, kde za znak = napíšou název funkce např. *modelNadrze*. Jako parametr funkce uvedou rozsah buněk s uloženými hodnotami vstupního signálu. Zadávaný vzorec nebude v tomto případě klasický, ale maticový. Je potřeba na listu vybrat souvislou oblast buněk o stejné velikosti shodné s rozsahem vstupních dat. Nad vyznačenou oblastí je nutné stisknout klávesu = a zadat název příslušné funkce např. *modelSystemu*. Do vybrané oblasti budou vypsány spočítané hodnoty po stisknutí kombinace kláves Ctrl, Shift, Enter. Tento postup je v Excelu zažit zejména pro maticové vzorce. Uživatel si proto postup s modelováním systému snadno osvojí.

Globálním využitím simulací by bylo univerzální pojetí modelování i se zadáním parametrů modelu a periody vzorkování, což je další možný přístup k řešení. Jednalo by se navíc o algoritmizaci teoretických postupů převodu diferenciální rovnice na stavový popis a určení jeho diskretní verze podle zdrojů [1], [3] až [7].

Při používání vzorců je patrná modularita řešení. Pokud vyučující naprogramuje více

vzorců, lze tak používat celou sadu funkcí kdekoli v rámci sešitu Excelu. Není cokoli vázáno na zvláštní spouštění maker přes tlačítka. Výstupní data mohou být dále snadno vizualizována pomocí grafů nebo může být proveden jejich export.

Studenti nemusí oplývat zkušenostmi z programování, což jim umožní jednoduše s modely pracovat. Znalosti programování v Microsoft Visual Basic for Application jsou potřebné v základní formě pouze u vyučujícího. Dále je uveden programový kód implementace numerické práce s modelem (10), jenž může být vyučujícím inspirací.

' sada potrebných maticových funkcí
Function scMt(matic1, matic2)

' součet matic

Dim soucet() As Double

ReDim soucet(UBound(matic1, 1),

UBound(matic1, 2))

For i = 1 To UBound(matic1, 1) Step 1

For j = 1 To UBound(matic1, 2) Step 1

soucet(i, j) = matic1(i, j) + matic2(i, j)

Next

Next

scMt = soucet

End Function

Function kMt(cislo, matic)

' vynasobeni matice cislem

For i = 1 To UBound(matic, 1) Step 1

For j = 1 To UBound(matic, 2) Step 1

matic(i, j) = cislo * matic(i, j)

Next

Next

kMt = matic

End Function

Function nasMt(matic1, matic2)

' nasobeni matic

Dim soucin() As Double

ReDim soucin(UBound(matic1, 1),

UBound(matic2, 2))

For i = 1 To UBound(soucin, 1) Step 1

For j = 1 To UBound(soucin, 2) Step 1

mezi = 0

For k = 1 To UBound(matic1, 2) Step 1

mezi = mezi + (matic1(i, k) *

matic2(k, j))

Next

soucin(i, j) = mezi

Next

Next

nasMt = soucin

End Function

Function umMt(matice, n)

' umocneni matice celým číslem

Dim vysledek() As Double

ReDim vysledek(UBound(matice, 1),

UBound(matice, 2))

vysledek = matice

If n > 1 Then

For i = 2 To n Step 1

vysledek = nasMt(vysledek, matice)

Next

End If

umMt = vysledek

End Function

' modelovani konkretniho systemu

Function modelSystemu(u As Range) As Variant

Dim y As Variant

Dim pom() As Double

y = u.Value 'vytvoreni pole

'(z pohledu dimenzi)

Dim F(2, 2) As Double 'stavove matice

Dim G(2, 1) As Double

Dim H(1, 2) As Double

Dim L(1, 1) As Double

Dim Stav(2, 1) As Double

F(1, 1) = 0.98

F(1, 2) = 0.24

F(2, 1) = -0.12

F(2, 2) = 0.93

G(1, 1) = 0.02

G(2, 1) = 0.15

H(1, 1) = 1

H(1, 2) = 0

For i = 1 To u.Count 'realizace vystupni

'rovnice

If i = 1 Then

Stav(1, 1) = 0 'pocatecni stav

Stav(2, 1) = 0

pom = nasMt(nasMt(H, F), Stav)

y(i, 1) = pom(1, 1)

End If

If i = 2 Then

pom = scMt(nasMt(nasMt(nasMt(H, F), F), Stav), kMt(u(1, 1), nasMt(H, G)))

y(i, 1) = pom(1, 1)

End If

If i > 2 Then

suma = 0

For R = 1 To (i - 1)

pom = kMt(u(1, 1), nasMt(nasMt(H, umMt(F, i - R - 1)), G))

suma = suma + pom(1, 1)

Next

pom = nasMt(nasMt(H, umMt(F, i)),

Stav)

y(i, 1) = pom(1, 1) + suma

End If

Next

modelSystemu = y 'navratove hodnoty

End Function

Uvedení funkcí pro práci s maticemi je čistě z komplexního důvodu, aby byla vyučujícím usnadněna celková implementace řešení. Při kopírování uvedeného kódu je nutné si dát pozor na rozdělení příkazů do více řádků, jenž zde bylo provedeno za účelem zkrácení textového rozsahu kódu. V programovacím prostředí je tedy nutné sloučení takto rozdělených řádků. Na schématu (Obr.2) je znázorněn způsob práce s vytvořeným rozhraním pro studenty.

vst. data	vypočítané
1	
1	
2	

vst. data	vypočítané
1	=modelSystemu(G31:G33)
1	
2	

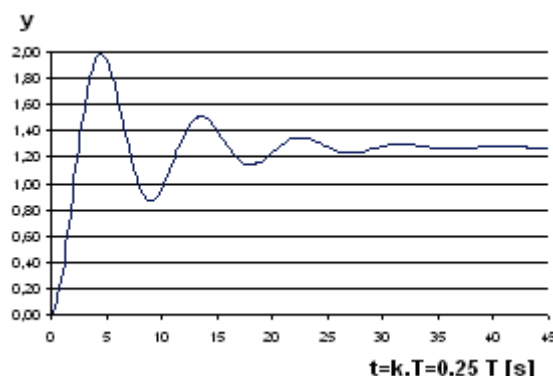
Ctrl+Shift+Enter

vst. data	vypočítané
1	0
1	0,02
2	0,1112

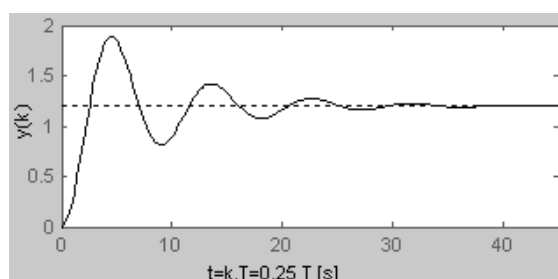
Obr 2: Postup výpočtu výstupního signálu konkrétního namodelovaného systému ve stylu vyhodnocení maticových vzorců v Excelu

5 Ověření platnosti navrhovaného přístupu k řešení modelování

Implementované řešení bylo verifikováno s výsledky spočítanými v nástroji MATLAB. V Excelu byla na vstup systému (10) zadána konstantní funkce s hodnotou 1. Výstupní veličina určená v Excelu byla srovnána s funkcí Step v softwaru MATLAB. Shodnost dosažených výsledků je patrná z grafů (Obr.3), (Obr.4). Čímž byla potvrzena správnost vytvořeného řešení.



Obr 3: Reakce modelu na jednotkovou konstantní funkci v Microsoft Excelu



Obr 4: Reakce modelu na jednotkový skok v prostředí MATLAB

6 Závěr

Příspěvek poukazuje na specifické výhody Microsoft Excelu pro výukové účely. Je předložen postup řešení numerických modelů a jeho aplikací, a to ve formě vytvoření vzorců v prostředí Excel, jenž vrátí výstupní diskretní posloupnost hodnot v reakci na zadaný vzorkovaný vstupní signál. Apriorním přínosem je poskytnutí výukové podpory studentům. Při práci s modelem získají kromě analýzy chování procesu též povědomí o možnostech simulace fyzikálních jevů. Může jim být též naznačen princip transformace vstupních funkcí na výstupní. Vyučující sice musí model sám naprogramovat, ale je mu zde poskytnut návod k jeho provedení. Množinu přínosů pro proces výuky komplementuje i inspirace samotné práce učitele studentům. Nebudou již vidět informatiku a fyziku odděleně, ale pochopí, že za pomoci aparátu matematiky je možné namodelovat záležitosti reálné praxe a informatika je pouze nástrojem, jenž vše aplikuje, ale s otevřenými možnostmi pro další potenciální bádání.

7 Poděkování

Článek byl uskutečněn za finanční podpory IGA Univerzity Tomáše Bati ve Zlíně, Fakulty aplikované informatiky číslo IGA/FAI/2012004.

8 Literatura

- [1] ŠTECHA, J. Teorie dynamických systémů. 1. vyd. Praha: České Vysoké Učení Technické v Praze, 2005. 248 s.
- [2] STREJC, V. Stavová teorie lineárního diskretního řízení. 1. vyd. Praha: Nakladatelství Československé akademie věd, 1978. 376 s.
- [3] SYSEL, M., BOBÁL, V. Moderní metody řízení – delta-modely. *Automa*. 2001, Praha, Ročník 2, Číslo 12, s. 17. ISSN 1210-9592.
- [4] BALÁTĚ, J. Automatické řízení. 1.vyd. Praha: Nakladatelství BEN- technická literatura, 2003. 664 s. ISBN 80-7300-020-2.
- [5] PROKOP, R., MATUŠŮ, R., PROKOPOVÁ, Z. Teorie automatického řízení - lineární spojité dynamické systémy. 1.vyd. Zlín: UTB ve Zlíně, Fakulta aplikované informatiky, 2006. 102 s. ISBN 80-7318-369-2.
- [6] VAŠEK, V. Teorie automatického řízení II. 1. vyd. Brno: Rektorát Vysokého učení technického v Brně, 1990. 139 s. ISBN 80-214-0115-X.
- [7] KUČERA, V. Analysis and Design of Discrete Linear Control Systems. 1.vyd. Praha: Nakladatelství Československé akademie věd, 1991. 480 s. ISBN 80-200-0252-9.
- [8] FORSTOVÁ, L. VBA Excel v příkladech. 1.vyd. Kralice na Hané: Computer Media, 2010. 120 s. ISBN 978-80-7402-042-1.
- [9] KRÁL, M. Excel VBA: výukový kurz. 1.vyd. Brno: Computer Press, 2010. 504 s. ISBN 978-80-251-2358-4.
- [10] BREDEN, M. Excel 2007 VBA: velká kniha řešení. 1.vyd. Brno: Computer Press, 2009. 685 s. ISBN 978-80-251-2698-1.

Ing. Tomáš Barot

Ústav řízení procesů

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

nám. T. G. Masaryka 5555

760 01, Zlín, ČR

Tel: +420 576 035 133

E-mail: barot@fai.utb.cz

Www pracoviště: <http://fai.utb.cz>