

# DATA INPUTS OF THE TKSL/C SIMULATION SYSTEM

Jiří KONVALINA, Master Degree Programme (5)  
Dept. of Intelligent Systems, FIT, BUT  
E-mail: xkonva02@stud.fit.vutbr.cz

Supervised by: Dr. Jiří Kunovský

## ABSTRACT

Taylor series computations are based on an automatic integration method order setting, i.e. using as many Taylor series terms for computing as needed to achieve the required accuracy. A new version of TKSL/C is experimentally implemented and computations with block input representations are analyzed.

## 1 ÚVOD

Metoda Taylorovy řady se velmi úspěšně uplatňuje při numerickém řešení obyčejných diferenciálních rovnic. Dosavadní implementace TKSL 386 již potvrdila řadu teoretických rozborů a byla úspěšně využita ve výuce. Nyní se řeší nový simulační systém TKSL/C jako knihovna v C++. Důležitou částí práce na tomto systému jsou výpočty s využitím blokového zadávání úloh. Implementují se postupně základní bloky a řeší se jejich návaznost na stávající TKSL/C.

## 2 NÁVRH A IMPLEMENTACE ZÁKLADNÍCH BLOKŮ SIMULAČNÍHO SYSTÉMU TKSL/C

Mezi základní bloky, které jsou pro systém TKSL/C implementovány, patří sumátor, integrátor, invertor, konstanta. Tyto bloky jsou navrženy jako nástavba systému TKSL/C. Cílem je pomocí blokové reprezentace zvýšení abstrakce problémů, které se ve spojitých simulacích řeší. Pro popis a charakteristické vlastnosti bloků je vytvořena dynamická knihovna.

### 2.1 DEFINICE JAZYKA DYNAMICKÉ KNIHOVNY

Jazyk je definován gramatikou, jejíž prepisovací pravidla jsou ve tvaru:

$S \rightarrow \text{HEAD}\{\text{str}\}.\text{ID}\{\text{num}\}.\text{FL}\{\text{str}\}.\text{EL}\{\text{num}\}A_1$   
 $A_1 \rightarrow \text{.BLOCK}\{A_2\}A_1 | \text{.BLOCK}\{A_2\}$   
 $A_2 \rightarrow \text{.NAME}\{\text{str}\}.\text{ID}\{\text{num}\}.\text{EQ}\{\text{str}\}.\text{IN}\{\text{num}\}.\text{OUT}\{\text{num}\}.\text{PC}\{\text{num}\}A_3$   
 $A_3 \rightarrow \text{.PRM}\{\text{num}\}\{\text{type}\}A_3 | \text{.SET}\{\% \text{num}\}\{\text{num}\}A_3 | \text{.SZ}\{\text{num num}\}A_4$   
 $A_4 \rightarrow \text{.DLG}\{A_5\}A_7 | A_7$   
 $A_5 \rightarrow \text{.ABOUT}\{\text{str}\}A_6$   
 $A_6 \rightarrow \text{.ITEM}\{\text{str}\}\{\% \text{num}\}A_6 | \text{.ITEM}\{\text{str}\}\{\% \text{num}\}$   
 $A_7 \rightarrow \text{.DRW}\{A_8\}$   
 $A_8 \rightarrow \text{.REC}\{A_{11} A_{11} A_{11}A_{11}\}A_9 | \text{.LINE}\{A_{11} A_{11} A_{11}A_{11}\}A_9 |$   
 $\quad \text{.CIRCLE}\{A_{11} A_{11} A_{11}A_{11}\}A_9 | \text{.TXT}\{A_{11} A_{11} A_{11}A_{11}\}A_9$   
 $A_9 \rightarrow A_8 | e$   
 $A_{10} \rightarrow \text{.USE}\{\% \text{num}\} | \text{str}$   
 $A_{11} \rightarrow \text{.MIDDLE} | \text{num}$ 

- $\text{num} - [0..9]^+ + ([0..9]^*.[0..9])$
- $\text{str}$  – libovolný řetězec znaků
- $\text{type}$  – typ proměrom[ $tS(\text{str}), tR(\text{num})$ ]

## 2.2 NAVRŽENÉ BLOKY SIMULAČNÍHO SYSTÉMU TKSL/C

- Bloky typu sumátor

Při návrhu sumátoru se vychází z obecné analýzy a vlastností, které blok sumátor má. Tyto vlastnosti jsou prostřednictvím jazyka dynamické knihovny implementovány do bloku sumátor. Nejprve byl vytvořen sumátor, který měl dva vstupy. Pomocí takto navrženého sumátoru jsou navrženy i vícevstupé sumátory. Liší se pouze v charakteristických vlastnostech jako jsou, počty vstupů a rozměry bloku. Jak je návrh a implementace dvouvstupého sumátoru realizována pomocí jazyka dynamické knihovna je ukázáno níže.

```

.BLOCK{                               //Sumátor, který má dva vstupy a jeden výstup
.NAME {Sum}                           //slouží pro název bloku
.ID {1}                                //jednoznačná identifikace bloku v knihovně
.EQ {#o1=#i1+#i2;}                   //popis chování bloku. o1 je výstup i1,i2 jsou vstupy
.IN {2}                                //počet vstupů bloku
.PC {0}                                //počet parametrů, které se dají v bloku nastavit
.OUT{1}                                //počet výstupů bloku
.SZ{30 50}                             //rozměry bloku jeho výška
.DRW{                                  //vykreslení bloku
.LINE{0 0 30 25}                      //vykreslení úseček
.LINE{30 25 0 50}
.LINE{0 50 0 0}
}
}

```

- Bloky typu integrátor

Podobně jako v bloku sumátor se využívá jazyk dynamické knihovny pro realizaci integrátorů. Návrh jednovstupého integrátoru je realizován triviálně. Problém však nastává při návrhu vícevstupých integrátorů. U vícevstupých integrátorů je nutné nejprve sečíst všechny jejich vstupní hodnoty a poté výslednou vstupní hodnotu integrovat. K realizaci součtu všech vstupních hodnot je možné použít sumátor, který má potřebný počet vstupů. Tento způsob realizace využívají některé komerční produkty. Je však patrné, že tímto způsobem se schéma zesložituje a proto je vhodné vytvoření vícevstupých integrátorů.

```

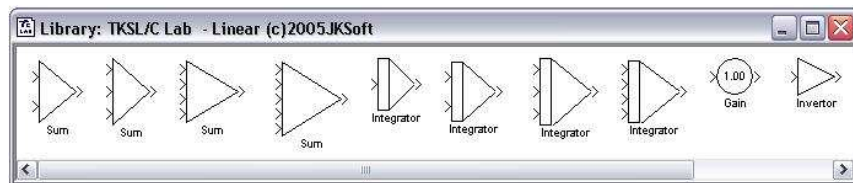
.BLOCK{                                     //Integrátor, který má jeden vstup a jeden výstup
.NAME {Integrator}                         //slouží pro název bloku
.ID {2}                                    //jedinečná identifikace bloku v knihovně
.EQ {#o1'=#i1;#o1(0)=%1;} //charakteristické vlastnost. %1 je parametr
.OUT{1}                                    //počet výstupů bloku
.IN {1}                                    //počet vstupů bloku
.PC{1}                                     //parametr bloku, zde počáteční podmínka
.PRM{1}{tR}                               //jakého typu je parametr
.SET{%1}{0}                               //přednastaven na hodnotu 0
.DLG{                                       //při kliknutí na blok se zobrazí dialog
.ABOUT{Continuous-time integration of the input signal}
.ITEM{Initial condition:}{%1} //možnost změnit parametr v dialogu
}
.SZ{30 40}                                //rozměr bloku, jeho výška
.DRW{                                       //vykreslení bloku
.REC{0 0 9 40}
.LINE{9 0 30 20}
.LINE{9 40 30 20}
}
}

```

- Bloky Invertor a Konstanta

Mezi základní bloky patří rovněž bloky invertor a konstanta. Návrh těchto bloků je velice jednoduchý, neboť z jejich obecné analýzy plyne, že mají jeden vstup a jeden výstup. Oba bloky jsou implementovány rovněž pomocí jazyka dynamické knihovny. Invertor má vlastnost takovou, že na výstup dá opačnou (inverzní) hodnotu vstupu. (.EQ {#o1=-#i1;}). Konstanta na svůj výstup vloží hodnotu vstupu, která je k-krát vynásobená. (.EQ {#o1=#i1\*%1;}, kde %1 je konstanta k).

- Jak navržené bloky vypadají je na obr. 1.



Obr. 1: Lineární bloky pro systém TKSL/C

### 3 ZÁVĚR

Navržený systém pro simulaci spojitých systémů je realizován pomocí výpočetního systému TKSL/C. Bloky, které jsou navrženy se dají uplatnit v mnoha odvětvích. Například v regulačních systémech lze pomocí základních bloků realizovat přenosové funkce. Cílem systému TKSL/C je vytvoření bloků, které přenosové funkce realizují pomocí jednoho bloku a tím uživateli zpříjemnit a zjednodušit práci. Již dnes se potvrdilo, že systém TKSL/C je v mnoha ohledech srovnatelný s existujícími systémy, které jsou nabízeny v komerční sféře.

### LITERATURA

- [1] Kunovský, J.: Modern Taylor series method. Habilitation work 1994 VUT Brno
- [2] Holub, O.: Systém TKSL, Diplomová práce FIT VUT Brno 2002