Lingvistické dynamické systémy a počítání se slovy pro modelování, simulaci a analýzu komplexních systémů

David Martinek podle článku kolektivu autorů Wang, Lin, Pu

10. února 2003

Abstrakt

Tato práce se zabývá teorií lingvistických dynamických systémů pro počítání se slovy. Slučuje procedury a koncepty z několika odlišných oblastí. Navrhovaný systém nám dovolí provádět globální dynamickou analýzu, systémový návrh a syntézu pro lingvistické dynamické systémy, které používají slova nebo lingvistické termíny pro výpočty. Tato teorie má důležitý potenciál pro modelování a analýzu systémů, kde model, cíle, řízení a zpětná vazba jsou primárně specifikovány pomocí slov nebo lingvistických termínů.

1 Úvod

Vývoj modelování a simulace na počítači a vývoj počítačových jazyků si jsou od počátku počítačového věku velmi blízké. Vývoj v jedné oblasti ovlivňuje vývoj v druhé a naopak. Od počátku ovlivňuje podobu programovacích jazyků i modelářských konceptů samotná architektura počítačů. Například „programovací jazyky“ pro analogové počítače měly podobu blokových schémat, ne nepodobných elektrickým obvodům. Na těchto strojích se řešily spojité simulační úlohy – protože se k tomu tato architektura nejvíce hodila.

S příchodem číslicových počítačů dostaly počítačové jazyky podobu, jakou známe dnes. Jejich koncepce se vesměs odvíjí od von Neumannova pojetí počítače[1]. Program v implerativním jazyce je složen z elementárních příkazů seřazených za sebou. Pro naše další úvahy je ovšem nejpodstatnější, že číslicový počítač používá jako základní datovou jednotku bity, respektive (na vyšší úrovni) celá čísla[2] Díky tomu, že má tato architektura teoreticky výpočetní sílu Turingova stroje, jsme schopni na ní řešit nejen diskrétní simulační úlohy, ale i úlohy spojité, kombinované či heterogenní[3]. Omezení dané celočíselnou architekturou jsme schopni

[1] Teď se nebudeme zabývat funkcionálními a logickými jazyky.

[2] Celá čísla se používají i pro reprezentaci nebo aproximaci reálných čísel.

[3] Heterogenní model používá i jiné koncepty než diskrétní či spojité, obvykle používá i koncepty z oblasti znalostního modelování.

eliminovat tím, že počítáme pouze s nějakou omezenou přesností (i když jsme schopni tuto přesnost vyčíslit či alespoň odhadnout).

Programovací jazyky byly původně určeny k tomu, aby přiblížily počítač více člověku. Programátor už nemusí zapojovat ručně stovky drátů nebo sapisovat instrukce ve strojovém kódu. Ovšem vývoj jde dál, dnes potřebujeme řešit stále složitější úlohy, které jsou v klasických programovacích jazycích typu Pascal, či C++ stále obtížněji řešitelné a jejich řešení se stále více přesouvá do oblasti známé jako umělá inteligence. Ideálním případem komunikace s počítačem by bylo použití lidského jazyka. Jakkoli se to může zdát pořád utopií z vědecko-fantastické literatury, už přes 40 let probíhá v této oblasti dosti intenzivní výzkum.

Na architekturu číslicového počítače je poměrně složité zavést systém, který by rozuměl lidské řeči. Teď nemám na mysli rozpoznávání slov, ale spíše ukládání a zpracovávání informací (znalostí), které jsou v lidské řeči obsaženy. V lidské mluvě je obsaženo velké množství nepřesných údajů, které navíc mohou záviset na okolnostech použití. Lidský jazyk je plný termínů jako *trochu, málo, skoro,* či *snad, možná,* atd. Tyto nepřesné údaje nejdou v číslicovém počítači, který počítá s absolutní přesností, nijak přímo uložit nebo zpracovat. V tradičním pojetí je modelování a simulace, stejně jako programování založeno na počítání s čísly a se symboly. Naproti tomu lidé pro počítání a rozhodování používají převážně slova a z premis popsaných přirozeným jazykem dochází k závěrům popsaným opět slovy. Pro překlenutí propasti mezi nepřesným lidským a absolutně přesným počítačovým jazykem vznikla matematická teorie fuzzy logiky, s níž v 60. letech 20. století přišel L.Zadeh [7]. Tato teorie se neustále vyvíjí a do budoucna bude velmi důležitá jak pro oblast modelování a simulace, tak i pro oblast programovacích jazyků.

Tato práce se věnuje hledání matematického rámce pro počítání se slovy pro popis, návrh a analýzu komplexních dynamických systémů. Pro konvenční dynamické systémy modelované pomocí diferenciálních či diferenčních rovnic bylo vyvinuto množství konceptů a metod pro systémovou analýzu i syntézu, přičemž tyto metody jsou dnes široce používány. Ovšem pro hodně velkých komplexních systémů je nepraktické nebo nemožné sestrojit konvenční matematický model kvůli přirozené vnitřní složitosti či neúplnosti informací (jak můžeme namodelovat chování trhu?). Dokonce i v případě, že byl nalezen odpovídající matematický model může být stále extrémně obtížné převést získaná data do formy akceptovatelné modelem a potom interpretovat výsledky analýzy a syntézy v lidsky srozumitelných termínech.

Naštěstí lidé mívají praxí „nakumulovanou“ zkušnost či znalost o práci takovýchto velkých komplexních systémů. Obecně je tento druh znalostí popsán přirozeným jazykem, to znamená pomocí slov nebo lingvistických termínů. Použitím těchto znalostí jsme schopni popsat, předvídat, řídit a vyhodnocovat chování i docela velkých komplexních systémů. Například ačkoli ve finanční oblasti jsou už řadu let studovány možnosti matematického předpovídání chování trhu, experti obvykle chování trhu odhadují sami a potom jsou schopni specifikovat svá rozhodnutí pomocí lingvistických termínů (tedy vlastními slovy). Aby byla analýza a syntéza takových systémů systematická, konzistentní a formální, potřebujeme vzít v úvahu speciální třídu dynamických systémů – lingvistické dynamické systémy (LDS). v LDS jsou problémy (zařízení), situace (stavy), plány (regulátory), měření (zpětná vazba), cíle a vyhodnocení (ukazatele výkonnosti) specifikovány v lingvistických termínech. Abychom mohli efektivně počítat s lidskými znalostmi, potřebujeme teorii pro modelování a syntézu lingvistických dynamických systémů.

V minulosti už bylo vynaloženo mnoho úsilí ve snaze dobrat se k tomuto cíli. Jako příklad můžeme zmínit znalostní systémy, expertní systémy, lingvistické struktury, vícehodnotovou logiku, fuzzy logiku, a mohli bychom jmenovat ještě další. Ačkoliv byly tyto metodologie úspěšně použity pro řešení mnoha konkrétních problémů, žádná z nich není schopná poskytnout teoretický rámec, jež by poskytoval koncepty a metody pro systémovou syntézu a analýzu analogické k těm, které jsou k dispozici pro konvenční dynamické systémy, jako je analýza stability či návrh řízení. Vezměme třeba problém vyhodnocování stability nebo návrh regulátoru pro systém popsaný množinou fuzzy pravidel typu IF–THEN. Prozatím pro to neexistoval žádný rozumný systematický přístup, například návrh takového regulátoru se nejčastěji dělá „ručně“ metodou pokus–omyl.

Hlavním zaměřením tohoto článku je představit teorii lingvistických dynamických systémů, která je výpočetně uskutečnitelná a která je schopná upotřebit koncepty a metody vyvinuté pro zkoumání konvenčních dynamických systémů, zvláště pak metody pro zkoumání stability, globální analýzu a pro návrh optimálních regulátorů.

Obrázek 1 ukazuje blokový diagram reprezentující LDS. Uvažujme proces vytváření regulačních zásahů na základě určitých problémů. v tomto systému jsou používána slova nebo lingvistické termíny pro velkou většinu celého výpočetního procesu, od specifikace po implementaci a vyhodnocení. Lingvistický dynamický model pro tento proces můžeme zkonstruovat tímto způsobem:

- *Slovně popsané problémy a situace:*

  Lingvistický popis plánů

- *Slovní vyjádření cíle:*

  Lingvistická cílová funkce

- *Slovní specifikace přístupu:*

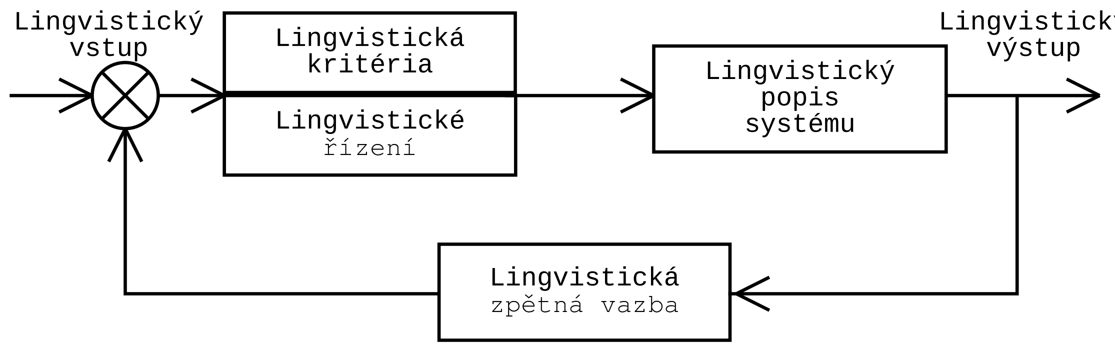
  Lingvistický návrh řízení

- *Slovní vyhodnocení procedur:*

  Lingvistická zpětná vazba

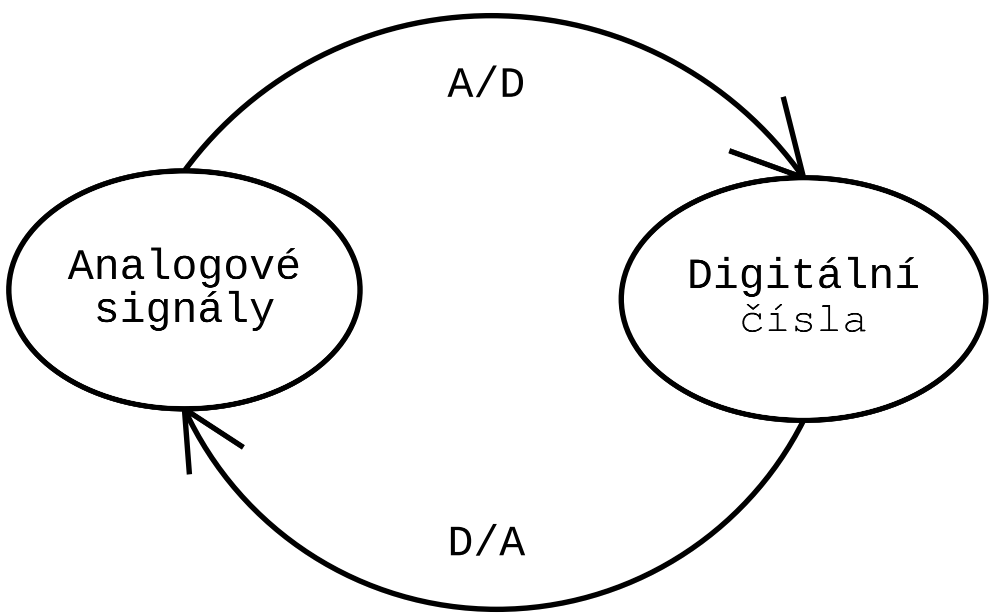
Tento model nám dovoluje provádět dynamickou simulaci pro různé přístupy na vysoké úrovni.

Pokud chceme na digitálním počítači pracovat s analogovými signály, pak potřebujeme analogově–digitální převodník (A/D). Když chceme, aby počítač komunikoval s okolím analogově, potřebujeme převodník digitálně–analogový (D/A). Takto můžeme digitální počítač zapojit do reálného fyzikálního systému (obrázek 2). Vývoj jde nezadržitelně vpřed a tak se neustále zvyšuje výkon počítačů. Až tento výkon dosáhne určité úrovně, budou počítače schopny zpracovávat jazykové informace podobně, jako jsou dnes schopny zpracovávat analogové signály. Když tuto analogii rozvedeme hlouběji, budeme potřebovat převodníky ze slov na čísla (word–number, W/N) a zpět z čísel na slova (number–word, N/W). Tyto převodníky



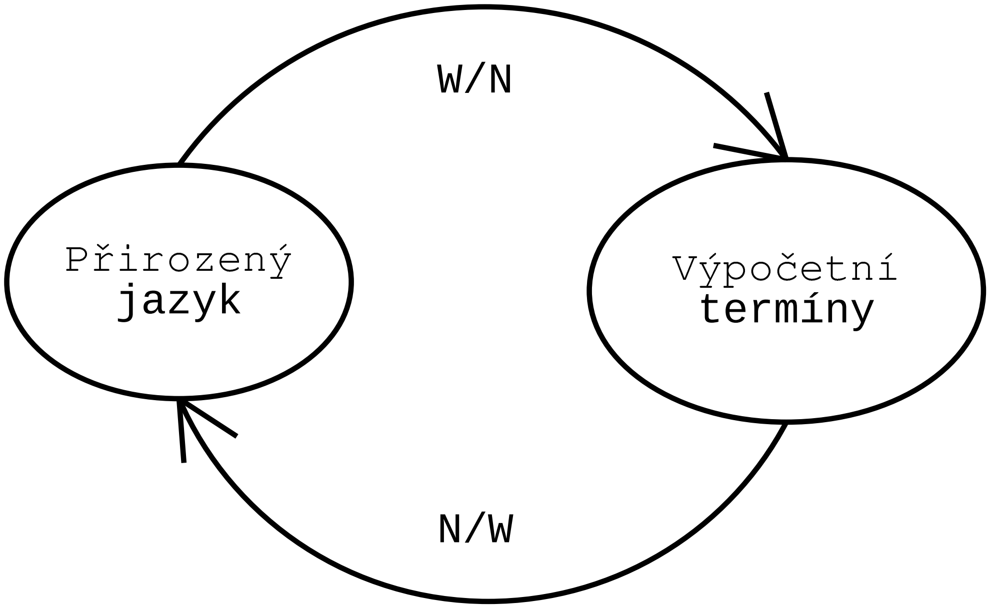
Obrázek 1: Blokový diagram reprezentující lingvistický dynamický systém.

(obrázek 3) budou sloužit jako rozhraní mezi člověkem a počítačem. Zatímco A/D a D/A převodníky jsou realizovány hardwarově, přičemž klíčem je přesnost, W/N a N/W převodníky jsou řešeny softwarově a klíčem je velikost slovníku.



Obrázek 2: Komunikace mezi fyzikálním systémem a počítačem: analogové signály se převádějí na čísla.

Naší základní myšlenkou zde bude reprezentovat LDS jako fuzzy dynamický systém. Dále budeme pracovat s Koskovou fuzzy hyperkostkou, kterou použijeme pro interpretaci LDS.



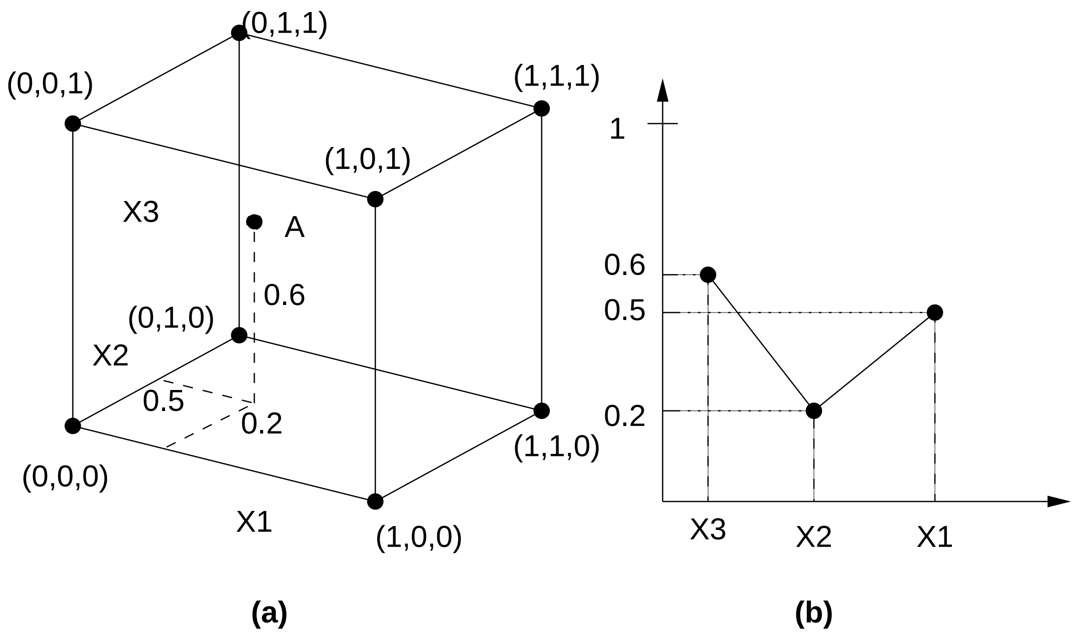
Obrázek 3: Komunikace mezi člověkem a počítačem: Slova a věty se převádějí na čísla.

2 Fuzzy množiny jako body v prostoru

Lofti Zadeh používal funkce příslušnosti k popisu fuzzy vztahů mezi entitou a množinou. Fuzzy množiny se pak staly ideálním prostředkem pro popis vágních výrazů v lidském jazyce. Ovšem další výzkum odhlalil, že reprezentace množiny pomocí dvourozměrných grafů přináší problémy. Kosko [6] oproti tomu navrhnul geometrickou reprezentaci fuzzy množin. Fuzzy množiny bere jako body v prostoru (obecně vícerozměrném). Tato interpretace umožňuje provádět analýzu stability LDS.

Abychom mohli dále pracovat s lingvistickými termíny, zavedeme si jako jednotku řečové informace termín *promluva*. Dále si zavedeme vektor $V_d = \{v_1, v_2, \ldots, v_n\}$, což je diskretizované univerzum promluvy. Funkcí příslušnosti pak je zobrazení $\mu_A : V_d \to [0,1]$, kde A je fuzzy množina. Tradičně je μ_A chápána jako dvourozměrný graf s definičním oborem V_d na jednorozměrné ose. Nová interpretace fuzzy množiny je založena na fuzzy potenční množině $F(2^{V_d})$, která obsahuje všechny fuzzy podmnožiny V_d a tvoří jednotkovou N-dimenzionální *fuzzy hyperkrychli* $I^N = [0,1]^N$. Za těchto podmínek můžeme fuzzy množinu považovat za bod nebo za vektor $X = (x_1, \ldots, x_n)^T \in F(2^{V_d})$, kde $x_i = \mu_A(v_i)$. Podle této interpretace vrcholy krychle I^N definují Booleovskou potenční množinu 2^{V_d}, která obsahuje všech 2^N ne-fuzzy množin z V_d. Takže fuzzy množiny vyplňují mezery ve 2^{V_d}, čímž vytvářejí celistvou fuzzy hyperkrychli $F(2^{V_d}) = I^N$. Na obrázku 4 je příklad fuzzy hyperkrychle pro $N = 3$, kde bod A (obrázek 4(a)) v krychli reprezentuje funkci příslušnosti (obrázek 4(b)).

Nyní můžeme na fuzzy zobrazení pohlížet jako na konvenční zobrazení na fuzzy hyperkrychli. To nám poskytne základ pro studium LDS na fuzzy hyperkrychlích. Problémy jako vzdálenost dvou fuzzy množin nebo jejich sousednost nyní můžeme řešit čistě geometricky.



Obrázek 4: Koskova fuzzy hyperkrychle a funkce příslušnosti.

Využijeme přitom znalosti vzorců pro vzdálenost bodů v N-rozměrném prostoru.

3 Formulace lingvistických dynamických systémů

Konvenční numerický dynamický systém můžeme považovat za časově závislé zobrazení z R^n do R^n a lingvistický dynamický systém za časově závislé zobrazení z I^N do I^N, přičemž N je mnohem větší než n (viz obrázek 5). Navíc použitím fuzzy množin jako lingvistických termínů, můžeme modelovat LDS jako fuzzy dynamický systém na fuzzy hyperkrychlích. Obecně může být LDS popsán takto:

Stavová rovnice:

$$X_{k+1} = F(X_k, U_k, k), \tag{1}$$

kde $F : I^n \times I^m \times Z^+ \to I^n$

Výstupní rovnice:

$$Y_k = H(X_k, k), \tag{2}$$

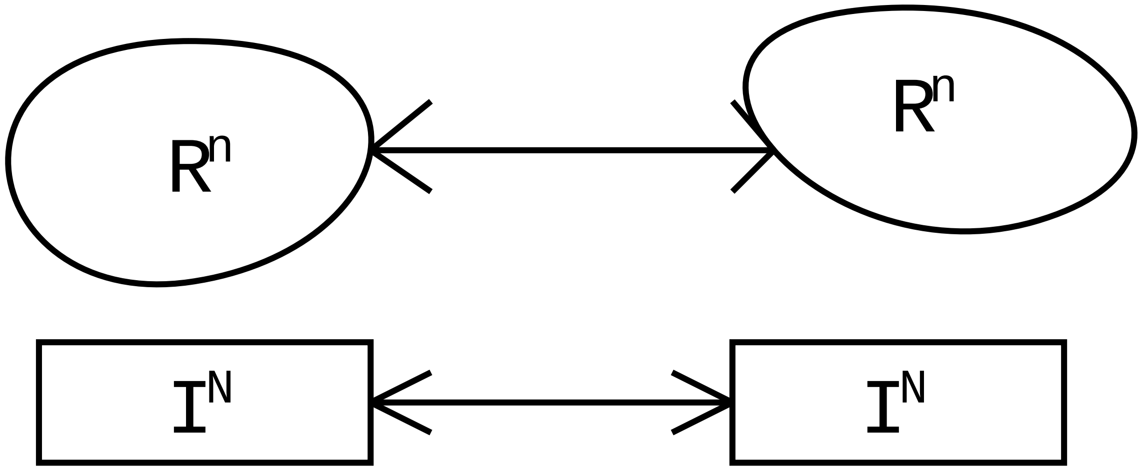
kde $H : I^n \times Z^+ \to I^n$

Zpětná vazba:

$$U_k = R(Y_k, V_k, k), \tag{3}$$

kde $R : I^p \times I^q \times Z^+ \to I^m$

Přičemž $Z^+ = \{1, 2, \ldots, K\}$, $X_k \in I^n$ je vektor reprezentující term pro stav systému, $Y_k \in I^p$ výstupní term, $V_k \in I^q$ vstupní term, $U_k \in I^m$ řídící term, k jednotka diskrétního času, a F, H, R jsou fuzzy operátory, jež definují systémové, výstupní a zpětnovazební zobrazení pro LDS.



Obrázek 5: Numerické versus lingvistické dynamické systémy.

Univerzum promluv každé proměnné tohoto systému je definováno jako

$$\begin{aligned} D_X &= \{x_1, K, x_n\}, \\ D_Y &= \{y_1, K, y_p\}, \\ D_U &= \{u_1, K, u_m\}, \\ D_V &= \{v_1, \Lambda, v_q\}. \end{aligned}$$

Odpovídající termy fuzzy množin jsou definovány jako

$$\begin{aligned} X &= \sum_{x_i \in D_X} \mu_X(x_i)/x_i, \\ Y &= \sum_{y_i \in D_Y} \mu_Y(y_i)/y_i, \\ U &= \sum_{u_i \in D_U} \mu_U(u_i)/u_i, \\ v &= \sum_{v_i \in D_V} \mu_V(v_i)/v_i. \end{aligned}$$

Různé systémy založené na fuzzy pravidlech jsou speciálními případy našeho formalismu. Začleněním výstupní rovnice a zpětné vazby do stavové rovnice, můžeme LDS popsat ve formě

$$X_{k+1} = F(X_k, V_k, k), k = 0, 1, 2, K$$

Když zde není žádný vstup, tj. $V_k = 0$, pak hovoříme o *autonomním LDS.*

4 Definice stability LDS

Využitím fuzzy krychlí můžeme téměř všechny koncepty vyvinuté pro konvenční dynamické systémy přímo zobecnit pro LDS. Následuje pář příkladů pro časově invariantní autonomní LDS $X_{k+1} = F(X_k)$.

Trajektorie LDS: Nechť F^k označuje zobrazení F aplikované k-krát. Zobrazení F^0 je identické zobrazení. Když začneme v X_0 LDS $X_{k+1} = F(X_k)$ generuje sekvenci termů $X_k = F^k(X_0), k = 0, 1, 2, K$. Tato sekvence termů se nazývá *trajektorie termů* nebo jednoduše *trajektorie* LDS s počátečním termem X_0.

Limitní term: Vezměme trajektorii $X_k = F^k(X), k = 0, 1, 2, K$. Term X^* se nazývá *limitní term* X v F, jestliže existuje sekvence kladných celých čísel n_k tak, že $n_k \to \infty$ a $F^{n_k}(X) \to X^*$ jako $k \to \infty$. Množina všech limitních termů X se značí $\Omega(X)$.

Vyvážený term: Term X se nazývá *vyvážený*, když $X = F(X)$.

P-K trajektorie: Periodická trajektorie LDS s periodou k je sekvence k odlišných termů $X_i, i = 1, 2, \Lambda, k$ takových, že

$$X_{i+1} = F^i(X_1), i = 1, 2, \Lambda, k-1, X_1 = F^k(X_1)$$

kterýkoliv term $X_i, i = 1, 2, \Lambda, k$ se nazývá periodický term s periodou k. Sekvenci (X_1, Λ, X_k) nazýváme *P-K trajektorie* a každý z jejích elementů *P-K term.*

Stabilita: Vyvážený term $X^* \in I^N$ je *stabilní*, když existuje takové $\varepsilon_0 > 0$, že platí: pro všechna $\varepsilon_1, 0 < \varepsilon_1 < \varepsilon_0$ existuje $\varepsilon > 0$ tak, že kdykoliv $\|X - X^*\|_W < \varepsilon$ a zároveň $X \in I^N$, pak $\|F^k(X) - X^*\| < \varepsilon_1$ pro všechna $k \geq 0$.

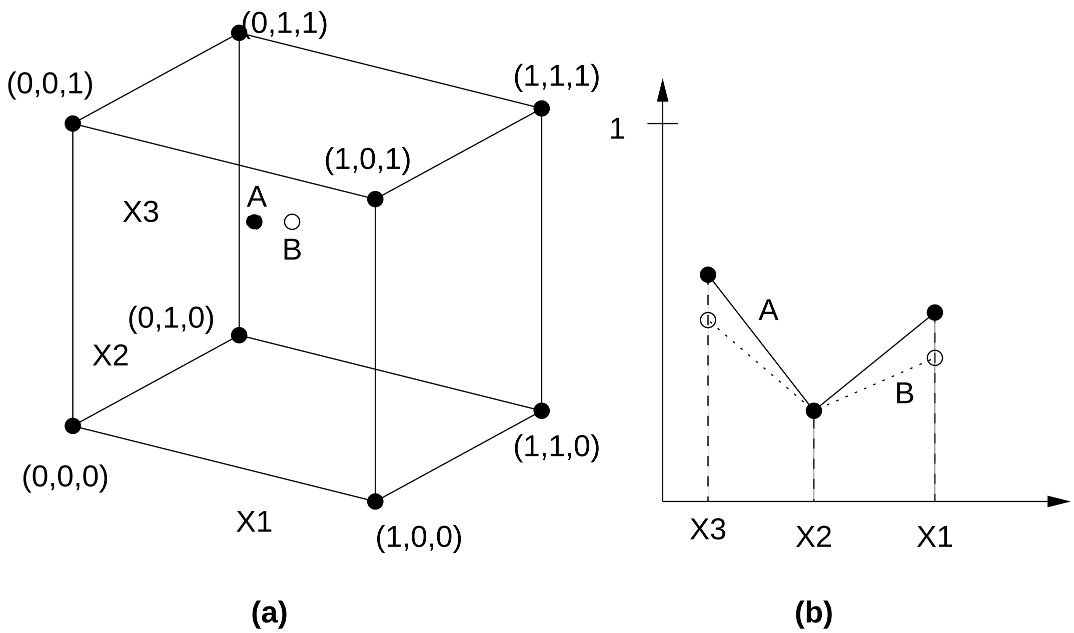
Operace $\|a - B\|_W$ je subjektivní vzdálenost dvou bodů ve fuzzy hyperkrychli. Počítá se jako klasická geometrická vzdálenost, ale můžeme do ní započítat míru subjektivity, kterou vyjadřuje matice W.

Asymptotická stabilita: Vyvážený term $X^* \in I^N$ je *asymptoticky stabilní*, když je stabilní a existuje $\varepsilon_0 > 0$ s touto vlastností: $\forall \varepsilon_1, 0 < \varepsilon_1 < \varepsilon_0$, jestliže $\|X - X^*\|_W < \varepsilon_1$ a zároveň $X \in I^N$, pak $F^k(X) \to X^X$ jako $k \to \infty$.

Globální asymptotická stabilita: Vyvážený term $X^* \in I^N$ je *globálně asymptoticky stabilní*, když je stabilní a pro vhodný počáteční term platí $X_0 \in I^N, F^k(X_0) \to X^*$ jako $k \to \infty$.

Mohlo by se zdát, že takto můžeme pracovat pouze s diskrétním univerzem, protože jinak bychom dostali hyperkrychle s nekonečným počtem dimenzí. v praxi to ovšem není problém, protože funkce příslušnosti jsou do značné míry subjektivní – nepotřebujeme je specifikovat s příliš velkou přesností, stačí nám znát jejich hodnotu pro pár typických bodů univerza. Čím větší přesnost budeme požadovat, tím větším rozměrem hyperkrychle se budeme muset zabývat.

Ukázali jsme si, že pro analýzu LDS můžeme použít metody vyvinuté pro konvenční numerické dynamické systémy. v praktických aplikacích ovšem někdy nemusí přímé použití konvenčních metod fungovat.

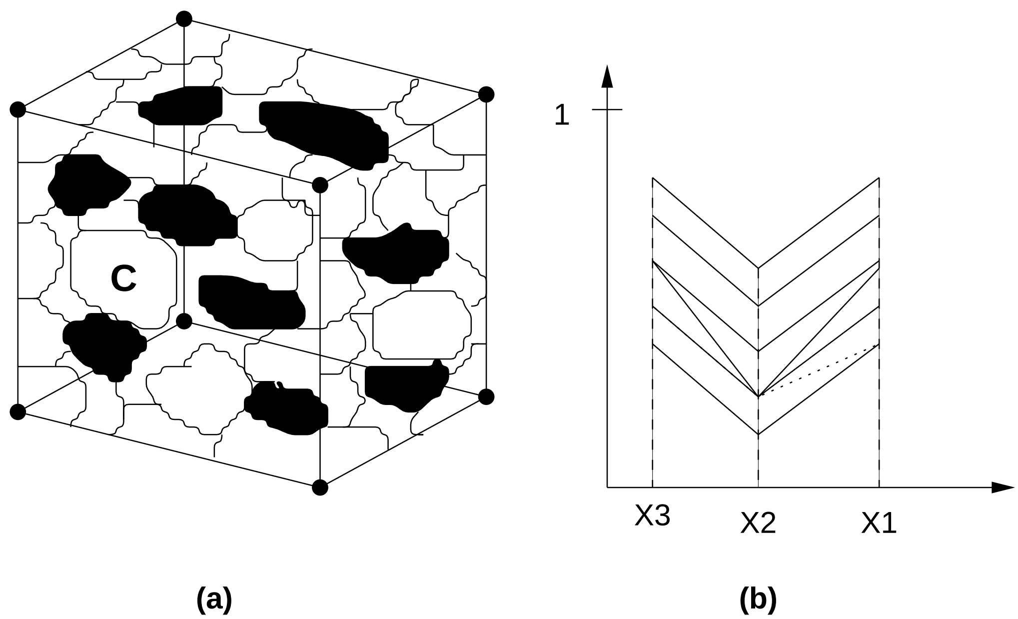


Obrázek 6: Bod a jeho soused ve fuzzy hyperkrychli.

Uvažujme například dva sousední body ve fuzzy hyperkrychli. LDS, který bude oscilovat mezi těmito body bude považován za nestabilní. Nicméně z praktického hlediska mohou být tyto dva body považovány za shodné, když například vyjadřují slova velmi podobného významu. Potom můžeme systém považovat za stabilní. Takovou situaci znázorňuje obrázek 6. Body A a B jsou navzájem různé, ale jejich funkce příslušnosti jsou si velmi podobné, proto je můžeme považovat za tentýž lingvistický term. To vede k úvaze, proč bychom nemohli považovat celé blízké okolí určitého bodu za jeden bod – jeden lingvistický term. Dalším argumentem pro uvažování tímto směrem je značná subjektivita fuzzy operátorů. Je známo, že pro jednu fuzzy operaci můžeme použít různé funkce, které dávají navzájem trochu odlišné výsledky, čili fuzzy zobrazení může mapovat ostrou hodnotu do různých bodů v závislosti na použité verzi operátoru. Tyto namapované body by měly ležet poměrně blízko sebe, takže celé blízké okolí můžeme považovat za jeden term. Tyto myšlenkové pochody vedly k zavedení celulárních struktur do fuzzy hyperkrychlí.

5 Celulární prostory ve fuzzy hyperkrychlích

Celulární struktura v prostoru může být zavedena několika způsoby. Obrázek 7 ukazuje obecnou celulární strukturu. Nejjednodušším způsobem je zavést rovnoměrné rozdělení prostoru na buňky. Základní ideou fuzzy buněk je považovat prostor ve fuzzy hyperkrychli ne za spojité kontinuum, ale za množinu buněk, kde je každá buňka brána jako samostatná entita.



Obrázek 7: Celulární struktura fuzzy hyperkrychle.

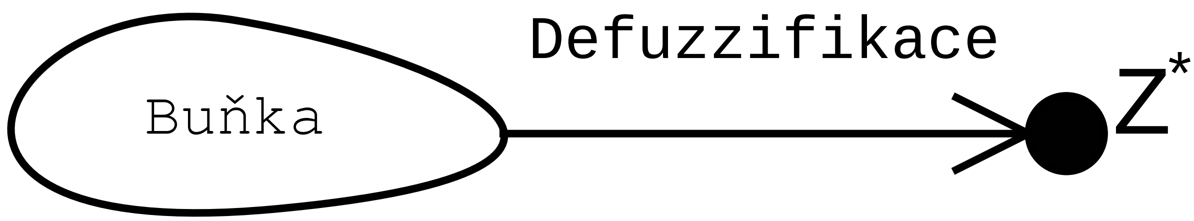
Nechť $x_i, i = 1, 2, \ldots, N$ je N komponent bodu ve fuzzy hyperkrychli I^N a ať osa proměnné x_i je rozdělena na větší množství stejných intervalů velikosti h_i. Interval z_i podle x_i-osy ať je definován tak, že obsahuje všechny x_i a splňuje

$$z_i - h_i/2 \leq x_i < z_i + h_i/2, 0 \leq x_i \leq 1$$

N-tice $(z_1, z_2, \ldots, z_N)$ se pak nazývá *buňkový fuzzy vektor* (nebo *fuzzy buňka*, či jenom *buňka*) a značí se Z. *Prostor stavů fuzzy buňky* S_C je kolekce všech takových buňkových vektorů. Jako reprezentativní bod celé buňky považujeme její střed, protože všechny body v buňce mají podobné funkce příslušnosti a považujeme je za jediný lingvistický term.

Poměrně zásadní otázkou při konstrukci celulárního prostoru pro specifikovaný počet termů N_t popisujících systém je, jak „optimálně“ tento počet termů nebo buněk rozmístit v celé hyperkrychli. Tato otázka je obzvláště významná pro hyperkrychle, které mají velký počet rozměrů. Jedním ze způsobů jak to udělat je distribuovat tyto termy tak rovnoměrně jak to jen jde. Toho se dá dosáhnout použitím ekvi-distribučních matric používaných v teorii numerické integrace pro vícerozměrné prostory.

Podobně jako se provádí defuzzifikace na fuzzy množinách, potřebujeme provádět defuzzifikaci na fuzzy buňkách. Pro zobecnění do více rozměrů je například vhodná metoda *těžiště*.



Obrázek 8: Defuzzifikace fuzzy buňky.

Při ní jde o to, najít geometrický střed plochy, nebo v našem případě buňky Z (viz obrázek 8). Příslušný vztah je zde

$$Z^* = \sum_{i=1}^{N} \alpha_i v_i, \alpha_i = \int_V \frac{x_i}{\sum_{j=1}^{N} x_j} dv \, / |v|$$

kde V je doména buňky Z.

6 Analýza LDS použitím cell-to-cell zobrazení

Protože jsme nahradili bodové funkce příslušnosti ve fuzzy hyperkrychlích buňkami, můžeme na ně použít analytické metody vyvinuté Hsuem pro nelineární diferenciální dynamické systémy [3, 4]. Tyto metody slouží k převodu LDS definovaném rovnicemi 1 až 3 na cell-to-cell[4] zobrazení,

$$Z_{k+1} = C_F(Z_k, Q_k, t_k),$$
$$P_k = C_H(Z_k, t_k), Q_k = C_R(P_k, W_k, t_k),$$

kde t_k je nový časový interval, $Z_k \in S_C^X, Q_k \in S_U^C, P_k \in S_C^Y, W_k \in S_C^V$ jsou buňky definované ve stavové, řídící, výstupní a vstupní fuzzy hyperkrychli a C_F, C_H a C_R jsou cell-to-cell zobrazení zkonstruované z F, H a R. Zkombinováním rovnic dostaneme tuto jedinou rovnici:

$$Z_{k+1} = C(Z_k, W_k, t_k),$$

Z_k, Z_{k+1} se nazývají *doménová buňka* a *obraz buňky.*

Všechny pojmy, které byly zavedeny pro fuzzy hyperkrychle, mohou být přímo použity i pro celulární prostory. Systémová analýza se, na rozdíl od hyperkrychlí, v celulárních prostorech stává výpočetně proveditelnou. Jakmile je známo zobrazení C, můžeme systematicky studovat globální dynamické vlastnosti LDS, tj. vyváženost buněk, P-K trajektorie a stabilitu. Pro účely numerické analýzy nyní zavedeme dvě další definice:

[4]Odpovídající český překlad by asi zněl *mezibuněčné zobrazení*, ale nezní to příliš matematicky, proto se budu raději držet originálního termínu.

R-krokově odstranitelná buňka: Buňka Z je *r-krokově odstranitelná* z P-K trajektorie, jestliže r je minimální celé číslo, takové že platí $C^r(Z) = Z_j$, kde Z_j je kterákoliv P-K buňka v P-K trajektorii.

R-kroková doména přitažlivosti: Množina buněk, která je z P-K trajektorie odstraněna v r nebo méně krocích, se nazývá *r-kroková doména přitažlivosti*. Když $r \to \infty$, dostáváme *totální doménu přitažlivosti* nebo jednoduše *doménu přitažlivosti* P-K trajektorie.

Jakmile je buněk konečně mnoho, každá buňka je buďto prvkem P-K trajektorie nebo r-krokové domény přitažlivosti P-K trajektorie. Hsuem a Guttaluem [4] byl popsán vyhledávací algoritmus pro analýzu autonomních dynamických systémů[5] používající zobrazení mezi buňkami. Tento algoritmus umí najít celkový počet M různých P-K trajektorií v systému a spojit každou buňku s jednou P-K trajektorií pro autonomní systém $Z_{k+1} = C(Z_k)$.

Předpokládejme, že máme vyčísleny všechny P-K trajektorie a máme je označeny jako $g = 1, 2, \ldots, M$. Pak nechť je $S(Z)$ počet kroků nutných pro dosažení P-K trajektorie g z buňky Z, $G(Z) = g$ ať je skupinové číslo dozažené P-K trajektorie, a $P(Z) = k$ je periodické číslo P-K trajektorie. Před startem algoritmu nejprve nastavíme $M = 0$ a rozdělíme buňky podle stavu vyhledávacího procesu do tří typů: *nezpracované buňky, právě zpracovávané buňky a zpracované buňky.* Skupinové číslo nezpracovaných buněk je nastaveno na nulu. Když se začne nezpracovaná buňka zpracovávat, nastaví se její skupinové číslo na -1. Po zpracování se buňce přiřadí odpovídající skupinové číslo a stává se zpracovanou buňkou.

Vezměme nezpracovanou buňku Z, průběh jejího zpracovávání generuje posloupnost obrazů buněk:

$$Z \to C(Z) \to C^2(Z) \to \ldots \to C^j(Z)$$

Vyhledávací algoritmus je popsán takto:

1. Nově generovaný obraz buňky $C^j(Z)$ je nezpracovaná buňka. v tomto případě změníme její skupinové číslo na -1, což znamená, že se stává zpracovávanou buňkou a pokračujeme na další obraz buňky $C^{j+1}(Z)$.

2. Nově generovaná buňka $C^j(Z)$ je zpracovávaná buňka s kladným skupinovým číslem. v tom případě je právě zpracovávaná sekvence zobrazena na buňku se známými vlastnostmi, tj. už byly nalezeny skupinové i periodické číslo i počet kroků. Je zřejmé, že všechny buňky v sekvenci budou mít skupinové i periodické číslo s $C^j(Z)$. Počet kroků ostatních buněk v sekvenci se spočte jako

   $$S(C^i(Z)) = S(C^j(Z)) + j - i, 0 \leq i \leq j - 1$$

   Všechny tyto buňky jsou pak zpracovány a označeny a vyhledávací proces pokračuje další nezpracovanou buňkou.

3. Nově generovaná buňka $C^j(Z)$ je právě zpracovávaná buňka. v tomto případě byla objevena nová periodicita, takže $M = M + 1$. Všechny buňky v sekvenci budou přiřazeny

[5] To znamená takových systémů, které nemají vstup.

do skupiny se skupinovým číslem $g = M$. Předpokládejme, že první znovuobjevená buňka bude $(i-1)$ buňkou v sekvenci, tj. $C^{l(Z)} = C^j(Z)$ pro minimum $l > i$. Potom $(l-i)$ je periodické číslo této nové periodické trajektorie a je přiřazeno každé buňce v sekvenci. Krokové číslo každé buňky se zpočte jako

$$S(C^k(Z)) = i - k, k = 0, 1, \dots, i-1,$$

$$S(C^k(Z)) = 0, k \geq i,$$

4. Když každá z předchozích situací skončí, algoritmus vezme další nezpracovanou buňku a začne nové hledání.

Tento proces se bude opakovat dokud nebudou zpracovány všechny buňky. Pak bude stanoveno globální dynamické chování celého systému. Výsledkem bude, že celý celulární prostor bude rozdělen na odlišné P-K trajektorie a jejich domény přitažlivosti. Každá buňka v celulárním prostoru patří k nějaké skupině. Krokové číslo buňky indikuje vzdálenost od její konvergující periodické trajektorie.

7 Návrh optimálního řízení

Problém návrhu optimálního řízení pro LDS je nalezení pravidel pro lingvistickou zpětnou vazbu, která optimalizuje index výkonnosti

$$J(U; X_0) = \sum_{i=0}^{T} \Phi(X_k, U_k, V_k, k), X_T \in \Gamma \subset I^n$$

kde Γ je podmnožina lingvistických cílů reprezentujících termy dosažitelné v T krocích, $\Phi : I^n \times I^m \times I^q \times Z^+$ je logický operátor jednokrokové lingvistické cenové funkce, I^l je fuzzy hyperkrychle univerza této funkce $D_\Phi = \{\varphi_1, \dots, \varphi_l\}$. Aby fuzzy sčítání dávalo smysl, můžeme předpokládat, že Φ vždy generuje fuzzy čísla. Nicméně víme, že toto není pro naši formulaci příliš důležité. Pro zjednodušení předpokládejme, že $Y = X$, tj. výstupní termy jsou tytéž jako termy stavové.

V celulárním prostoru může být jednokroková cenová funkce aproximována cell-to-cell zobrazením

$$\Phi_k = C_\Phi(Z_k, Q_k, W_k, t_k), Z_T \in S_\Gamma \subset S_C^X$$

kde S_Γ je celulární podprostor cílového podprostoru a Φ_k je buňka v I^l. Použitím defuzzifikační metody zavedené v sekci 5, můžeme defuzzifikovat Φ_k tak, že dostaneme pro buňku Φ_k numerickou hodnotu $\gamma_k = DF(\Phi_k)$.

Jednoduše, pomocí cell-to-cell zobrazení se návrh optimálního regulátoru pro LDS stává standardním dynamickým programovacím problémem s diskrétními stavovými a kontrolními proměnnými. Díky tomu můžeme použít všechny metody vyvinuté pro dynamické programování.

Obecný vyhledávací algoritmus pro optimální lingvistické řízení byl navržen Puem. v této metodě je jednokroková cena kvantifikovaná do konečných úrovní $\beta_1\Delta < \beta_2\Delta < \ldots < \beta_L$, kde β_i jsou celá čísla a Δ je kvantifikační jednotka. Akumulovaná cena se vyčíslí

$$J = \Delta \sum_{i=1}^{L} k_i \beta_i$$

kde k_i jsou celá čísla.

Vyhledávací algoritmus může být jednoduše popsán takto:

1. Nastav $S_0 = S_\Gamma$ a $i = 1$. Označ všechny buňky v S_Γ jako zpracované a všechny ostatní buňky jako nezpracované.
2. Vyhledávej mezi všemi nezpracovanými buňkami buňku, která může být zobrazena do $S_{i-1} Y \Lambda Y S_0$ s i-tou akumulovanou cenovou úrovní. Ať S_i je množina všech nově naležených buněk. Označ všechny buňky v S_i jako zpracované.
3. Jestliže už byly označeny všechny buňky, které nás zajímají, označeny jako zpracované, skonči. v opačném případě nastav $i = i + 1$ a vrať se zpět na krok 2.

Když je algoritmus kompletní, máme k dispozici tabulku optimálního řízení, která poskytuje optimální řídící akce pro dosažení specifikovaného cíle S_Γ pro každou buňku.

8 Závěr

V této práci je načrtnuta výpočetní terorie pro lingvistické dynamické systémy. Tato teorie je spojením procedur a konceptů z různých oblastí: geometrická teorie fuzzy množin, cell-to-cell zobrazení v nelineární analýze, ekvi-distribuční matice v teorii čísel a dynamické programování v teorii optimálního řízení. Tato teorie nám umožňuje numericky spojit globální dynamickou analýzu, systémový návrh a syntézu pro lingvistické dynamické systémy založené na konceptech a metodách vyvinutých pro konvenční dynamické systémy.

Reference

[1] F-Y Wang, Y.Lin, J.B.Pu: *Linguistic Dynamic Systems and Computing with Words for Modeling, Simulation, and Analysis of Complex Systems*. Discrete event modeling and simulation technologies, chap 5., pp75–92, Springer–Verlag, New York, 2000.

[2] H.S.Sarjoughian, F.E.Cellier: *Discrete event modeling and simulation technologies – A Tapestry of Systems and AI-Based Theories and Methodologies*. Springer–Verlag, New York, 2000.

[3] C.S.Hsu: *Cell-to-cell mapping*. Springer-Verlag, New York, 1987.

[4] C.S.Hsu, R.S.Guttalu: *An Unravelling Algorithm for Global Analysis of Dynamic Systems: An Application of Cell-to-cell Mapping*. ASME J. Appl. Mechanics, vol. 47, pp.940–948, 1985.

[5] L.K.Hua, Y.Wang: *Applications of Number Theory to Numerical Analysis*. Springer–Verlag, New York, 1981.

[6] B.Kosko: *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Englewood Cliffs, NJ: Prentice Hall, 1992.

[7] L.A.Zadeh: *Fuzzy Sets, Information and Control*. vol. 12, pp338–353, 1965.