

USING OF ARTIFICIAL INTELLIGENCE METHODS FOR CONTROL OF MANIPULATORS WITH ROTARY ARM

Robert Grepl*, Čestmír Ondrušek†

Summary:

This paper deals with using of evolution algorithms for optimization of control of systems, where three degrees of freedom are controlled by only one action quantity. There is the differential evolution used for optimization. Performance of this method is compared with genetic algorithm. It is possible to generalize this approach for solution of similar problems of various mechanisms and systems.

Key words: control of mechanisms, optimization, using of differential evolution

Úvod

Při návrhu počítačového řízení je cílem dosažení požadovaného chování řízené technické soustavy. V případě, že je možné v každém stupni volnosti působit příslušnou akční silou, např. všechny pohyblivé klouby manipulátoru robota jsou osazeny servopohony, je problémem kinematika pohybu, dynamika pak ovlivňuje dimenzování pohonů.

Existují ale také soustavy, ve kterých není možné všechny stupně volnosti přímo ovládat. Příkladem může být stabilizace kývání břemene mostového jeřábu. Jednou veličinou - polohou vozíku - v tomto případě ovlivňujeme dva stupně volnosti - výkyv břemene a vlastní polohu vozíku. Potom nelze realizovat optimální (požadované) řízení obou stupňů volnosti. Je však možné řídit soustavu tzv. kvazioptimálně, kdy dosahujeme žádaného stavu obou řízených veličin pouze v jistých diskrétních časových okamžicích [1], což ovšem může z hlediska praktické funkce zařízení zcela postačovat. V případě mostového jeřábu jsou těmito diskrétními body počáteční a koncový stav, kdy bude rychlost jeřábu a výchylka břemene nulová. Přitom průběh kývání při přejíždění do koncové polohy není z hlediska požadavku na řízení důležitý.

Další možností jsou soustavy, ve kterých jsou dva a více stupňů volnosti řízeny jednou veličinou. Pak ovšem nelze realizovat ani kvazioptimální řízení způsobem popsáným v [1]. Příkladem je otočný manipulátor na obr. 1. Jednou akční veličinou - otáčením okolo osy z - je ovlivňováno kývání kyvadla v prostoru, tedy dva stupně volnosti. Cílové chování může být stanoveno podobně jako u řízení mostového jeřábu. Budeme požadovat minimální kývání po pootočení ramene z počáteční polohy, kdy je kyvadlo v klidu, do koncové polohy.

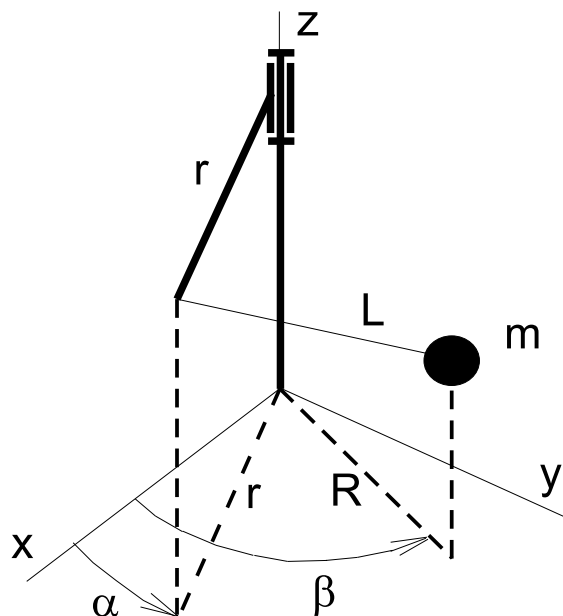
Tento příspěvek se zabývá využitím evolučních algoritmů k řešení popsaného problému, konkrétně byla použita metoda diferenciální evoluce. Nepřímou úlohu dynamiky (požadujeme jisté výsledné chování) tak převádíme na mnohonásobné opakování přímé úlohy.

*Ing. Robert Grepl, Centrum pro mechatroniku, společné pracoviště ÚT AV ČR a FSI VUT v Brně, Technická 2, 616 69, Brno; tel. +420 5 4114 2757, e-mail: grepl@uvec.fee.vutbr.cz

†Doc. Ing. Čestmír Ondrušek, CSc., UVEE FEI VUT v Brně, Technická 8, 616 69, Brno; 4114 2364, e-mail: ondrusek@uvec.fee.vutbr.cz

Vytvoření dynamického modelu soustavy

Geometrie otočného manipulátoru je na obr.1. Uvažujme tyto zjednodušující předpoklady: rameno



Obrázek 1: Geometrie modelu soustavy

manipulátoru budeme vyšetřovat pouze z hlediska kinematiky, nezajímá nás tedy dynamika pohonu; hmotu břemene soustředíme do hmotného bodu m ; lano L považujeme za dokonale tuhé, nehmotné a konstantní délky; neuvažujeme odpory prostředí. Nyní definujeme zobecněné souřadnice α , β , r a R podle obrázku a můžeme přistoupit k vyjádření kinetické a potenciální energie soustavy. Po úpravě obdržíme:

$$Ek = \frac{1}{2}m_2((\dot{R}\cos(\beta) - R\dot{\beta}\sin(\beta))^2 + (\dot{R}\sin(\beta) + R\dot{\beta}\cos(\beta))^2 + \quad (1)$$

$$\frac{(2r\dot{R}\cos(\alpha - \beta) + 2rR\sin(\alpha - \beta)(-\dot{\alpha} + \dot{\beta}) - 2R\dot{R})^2}{4(L^2 - r^2 + 2rR\cos(\alpha - \beta) - R^2)})$$

$$Ep = -9.81m_2(L^2 - r^2 + 2rR\cos(\alpha - \beta) - R^2)^{\frac{1}{2}} \quad (2)$$

Z hlediska sledování chování kyvadla jako subsoustavy je výhodné provést následující substituci:

$$\delta = \beta - \alpha \quad (3)$$

$$w = R - r \quad (4)$$

Pomocí Lagrangeových rovnic II. druhu pak sestavíme dynamický model. Výsledné rovnice jsou velice obsáhlé a proto uvedeme pouze maticový zápis soustavy ODE. Pro řešení rovnic v MATLABu je nutné provést převod n diferenc. rovnic druhého řádu na $2n$ rovnic prvního řádu substitucí $y_1 = \delta; y_2 = \dot{\delta}; y_3 = w; y_4 = \dot{w}$.

$$\mathbf{M}\dot{\mathbf{y}} = \mathbf{C} \quad (5)$$

$$\mathbf{M} = \begin{bmatrix} M_{1,1} & M_{1,2} & 0 & 0 \\ M_{2,1} & M_{2,2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

$$\dot{\mathbf{y}} = [\dot{y}_1 \cdots \dot{y}_4]^T \quad (7)$$

$$\mathbf{C} = [C_1, C_2, y_1, y_2]^T \quad (8)$$

Obdrželi jsme soustavu nelineárních diferenciálních rovnic, kterou budeme řešit v MATLABu pomocí solveru `ode45`. Výpočet jedné simulace trvá na běžném PC řádově sekundy až desítky sekund, což je přijatelné pro optimalizační výpočet.

Optimalizační výpočty

Nejprve popíšeme základní myšlenku optimalizace pomocí evolučních algoritmů. Problémem je nalezení vhodného časového průběhu otáčení ramene manipulátoru, tedy funkce $\alpha = \alpha(t)$, takovou, aby na konci pohybu byla zbytková výchylka kyvadla minimální. Hodnotící funkce daného řešení má tak jako argument tvar funkce α a na výstupu velikost zbytkové výchylky. Pomocí evoluční metody pak měníme tvar funkce α s cílem minimalizovat zbytkovou výchylku.

Otázkou je, jakým způsobem reprezentovat pomocí parametrů evolučního algoritmu (genů daného jedince v populaci) tvar funkce α . Nabízí se využití polynomů nebo lineární kombinace různých funkcí, možné by také bylo určení funkce pomocí genetického programování. Pokusy s polynomy a různými funkcemi se ale ukázaly jako neefektivní, proto jsme použili následující jednoduchou metodu.

Zvolíme určitý (malý) počet parametrů (genů) n . Jejich reálné hodnoty představují velikost α postupně v n časových okamžicích. Mezi těmito body pak proložíme vhodnou funkci. Vzhledem k požadavku jednoduchosti jsme zvolili tuto funkci:

$$\alpha_i(t) = \frac{A_i}{2} \left(1 - \cos\left(\frac{\pi}{B_n}(t - t_n)\right) \right) + C_n \quad (9)$$

kde $\alpha_i(t)$ je průběh funkce v i -tém intervalu (mezi n -tým a $(n+1)$ -tým bodem); $A_i = \alpha(t_{n+1}) - \alpha(t_n)$; $B_i = t_{n+1} - t_n$; $C_i = \alpha(t_n)$. Jistou nevýhodou z hlediska praktické aplikace je diskontinuita v průběhu zrychlení, ovšem hlavní požadavek na nulovou rychlost a zrychlení v koncových bodech je splněn.

Popis algoritmu diferenciální evoluce (DE)

Diferenciální evoluce [2] je poměrně novou metodou, která se však svým principem velice podobá známému genetickému algoritmu (GA). Oproti němu je výrazně jednodušší a tím i rychlejší v jednotlivých svých iteracích. Co se týče celkové rychlosti nalezení globálního minima, je jakékoli srovnávání diskutabilní, patrně neexistuje metoda, která by obecně dokázala, který z evolučních algoritmů bude rychleji konvergovat, vždy záleží na konkrétní úloze a nastavení parametrů algoritmů. Podle [2] bývá DE úspěšnější v nalezení globálního oproti lokálnímu minimu. V každém případě se v naší aplikaci DE osvědčila lépe než GAOT¹.

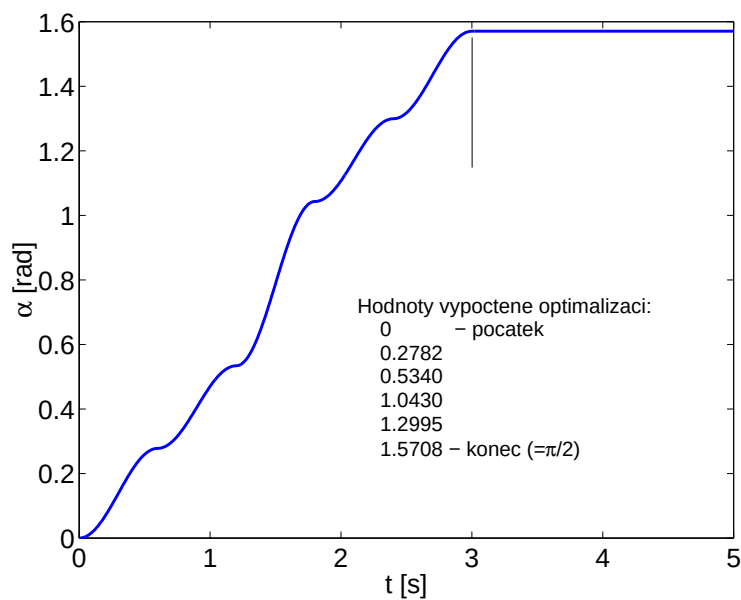
Poměrně podrobný popis DE je v [2], podobně jako v GA vytvoříme populaci jako množinu jedinců a náhodně vygenerujeme hodnoty jejich genů. Samotná evoluce pak probíhá v generacích a je řízena dvěma parametry - váhovou konstantou a prahem křížení. Podobně jako u GA je možné metodu vylepšovat průběžnou změnou parametrů evoluce během výpočtu (METADE).

Výsledky optimalizačního výpočtu

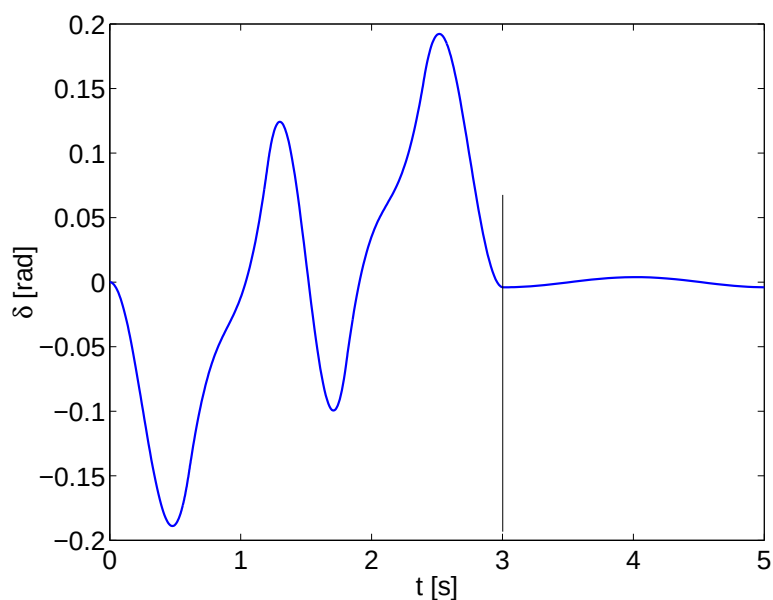
Parametry modelu a nastavení algoritmu DE byly následující: délka otočného ramene $r = 0.6 \text{ m}$; délka lana $L = 1 \text{ m}$; hmotnost břemene $m_2 = 1.6 \text{ kg}$; požadované natočení $\alpha_K = 90^\circ = 1.57 \text{ rad}$; požadovaný čas natočení $t_K = 3 \text{ s}$; počáteční podmínky $\alpha_0, w_0, \delta_0, \dot{\alpha}_0, \dot{w}_0, \dot{\delta}_0 = 0$; povolený rozsah $\alpha = (-0.2; 2) \text{ rad}$; počet genů v jedinci = 4; práh křížení DE = 0.5 váhová konstanta DE = 1; počet jedinců v populaci = 10.

¹GAOT - Genetic Algorithm Toolbox pro MATLAB

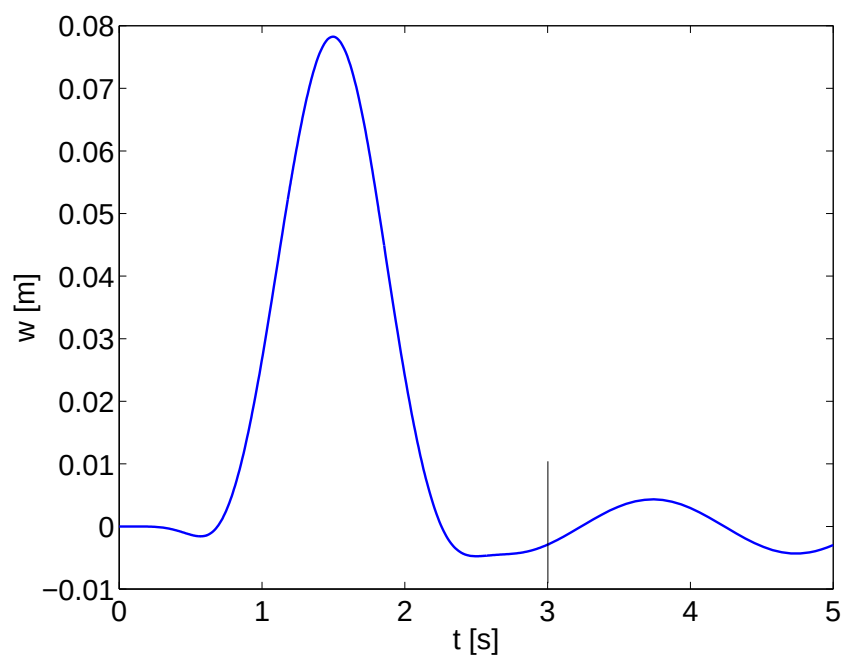
Výpočet trval na počítači Pentium III řádově desítky hodin. Výsledky optimalizace jsou na obr. 2, 3, 4 a 5. Simulace byla prováděna vždy 3 s kdy probíhalo vlastní otáčení mechanismu a další 2 s, kdy jsme sledovali zbytkovou výchylku, určili její maximální hodnotu a tuto prohlásili za max. chybu. V našem případě bylo dosaženo hodnoty chyby 0.004 m. Z obrázku 4 je patrné, že zbytkové kývání probíhá především v ose w.



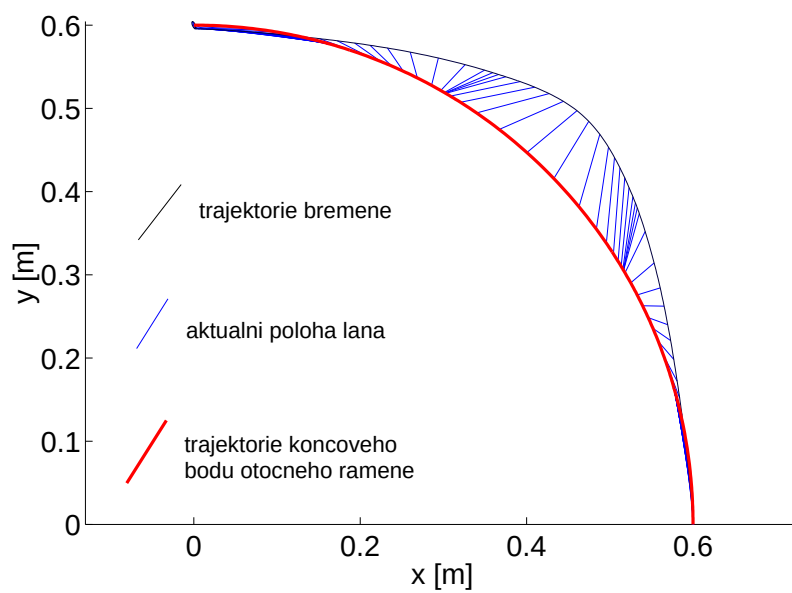
Obrázek 2: Závislost otočení ramene α na čase



Obrázek 3: Závislost výchylky δ na čase



Obrázek 4: Závislost výchylky w na čase



Obrázek 5: Pohyb mechanismu při otáčení při pohledu shora

Závěr

Výše popsaným způsobem výpočtu dráhy otočného ramene pomocí DE se podařilo dosáhnout uspokojivých výsledků. Z obrázků je patrné, že zbytková výchylka je velmi malá a z hlediska funkčnosti zařízení zcela zanedbatelná. Obecně je možné výše popsaný algoritmus použít na celé řadě jiných principiálně podobných problémů.

Z hlediska možného praktického uplatnění je ale překážkou potřebný výpočtový čas optimalizace. Řešením by mohlo být provedení velkého množství optimalizačních výpočtů předem, pro různé počáteční podmínky, a při řízení pak z těchto dat vhodně interpolovat parametry pro konkrétní požadované podmínky. Možná by byla aplikace ANN (umělých neuronových sítí) pro tuto aproximaci, podobně jako v [3]. V této souvislosti je aktuální otázka citlivosti celého algoritmu na nepřesnost jednotlivých parametrů. Dále by bylo nutné do optimalizačního výpočtu zahrnout vlastnosti zvoleného pohonu, např. kontrolovat nepřekročení maximální dovolené rychlosti, zrychlení a momentu.

Poděkování

Tento příspěvek vznikl za podpory projektů GA ČR 102/00/1586 a CEZ: J22/98:262200010.

References

- [1] Valášek, M.; Miláček, S.; Dedouch, K. *Position and Velocity Control of gantry crane*. In: Proc. of Mechatronics 96, Guimaraes 1996, Vol.1, pp.203-208
- [2] Zelinka, I. *Aplikovaná informatika aneb úvod do fraktální geometrie, buněčných automatů*. Ediční středisko FT, (1997).
- [3] Grepl, R., Veselý, M., Ondrůšek, Č. *Position control of gantry crane in real time*. Kolokvium Diagnostika a aktivní řízení, Třešť (2000).