

НЕВРОННО-РАЗМИТО ПРЕДСКАЗВАЩО УПРАВЛЕНИЕ НА ЛАБОРАТОРЕН МОДЕЛ НА ДВУРОТОРЕН ХЕЛИКОПТЕР

FUZZY-NEURAL MODEL PREDICTIVE CONTROL OF LABORATORY MODEL OF TWO-ROTOR HELICOPTER

S. Ahmed*, M. Petrov**, A. Ichtev*

* Technical University of Sofia, Systems and Control Department, Sofia, Bulgaria

** Technical University of Sofia, branch Plovdiv, Control Systems Department, Plovdiv, Bulgaria

e-mail: sevil.ahmed@gmail.com, mpetrov@tu-plovdiv.bg, kish@tu-sofia.bg, ichtev@tu-sofia.bg

Abstract: An application of Model Predictive Controller (MPC) on the nonlinear laboratory model is presented in this paper. Nonlinear Model Predictive Control (NMPC) is studied as a more applicable approach for optimal control of nonlinear plants in industry in our days. A two-rotor aerodynamical laboratory model is studied. The Takagi-Sugeno type fuzzy neural network is used to model the nonlinear plant. The model is embedded into a MPC scheme and ensures solving the optimization procedure of MPC as a quadratic programming problem with constraints.

Key words: Optimization, Nonlinear Predictive Control, Fuzzy-Neural Model, Nonlinear Systems

ВЪВЕДЕНИЕ

Моделното предсказващо управление (МПУ) е един от най-широко прилаганите подходи за адаптивно управление на динамични системи в индустрията. Основната идея при МПУ е да се реши предварително дефинирана оптимизационна задача с цел намиране на вектор на управленията, който минимизира целевия критерий зададен в оптимизационната процедура за целия хоризонт на предсказване [2,3]. Първоначално, МПУ е предназначено и се прилага само за линейни обекти и процеси [1]. В последствие то е разширено и доразвито за нелинейни приложения, в съответствие с характеристиките и нуждите на повечето обекти и процеси в индустрията.

В практиката преобладават широк клас промишлени процеси, които са съществено нелинейни и те са обект на нелинейното моделно предсказващо управление (НМПУ). В последните години редица изследователи публикуват разработки и алгоритми за НМПУ [2], в които се използват различни подходи за нелинейно моделиране. Някои от тези модели се базират на класическата теория за нелинейно управление, други използват емпирични данни с приложение на невронни мрежи и размита логика за построяване на нелинейни модели.

Точният модел на изследвания обект или процес определя в голяма степен ефективността и работоспособността на приложението използващо подхода на МПУ. Съществуват различни модификации на МПУ различаващи се по вида на използвания предсказващ модел – адаптивни модели [4], невронни модели [3] и др. Размитите и невронните модели се отнасят към широко известния и прилаган подход на „черната кутия“.

Променящата се динамика и наличието на смущения от различно естество при обектите налагат и използването

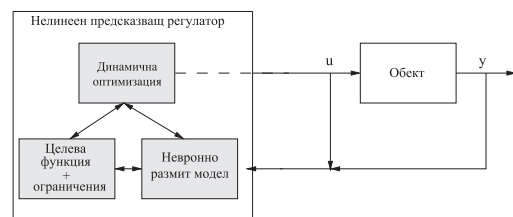
на динамични модели. Невронните и размитите модели удовлетворяват тази необходимост за адаптация в реално време. Невронно-размитият подход от своя страна съчетава свойствата и на двата типа модели и се превръща в мощен апроксиматор в реално време за обекти и процеси с нелинейни характеристики.

Другият основен етап при НМПУ е изчисляването на множеството от бъдещите управляващи въздействия. Това е етапът на решаване на оптимизационната задача на НМПУ. Едно от най-големите предимства на МПУ е възможността за въвеждане на ограничения [1].

В представената разработка се използва невронно-размит модел на Такаги-Сугено [6] в пространството на състоянията с приложение за предсказващо управление. Предложеният подход за невронно-размит НМПУ е представен чрез система за управление на двуроторен аеродинамичен модел с две степени на свобода. Изследванията са проведени в симулационната среда на MATLAB/Simulink.

НЕВРОННО-РАЗМИТ ПРЕДСКАЗВАЩ МОДЕЛ

Основният подход за НМПУ с използване на невронно-размит предсказващ модел на Такаги-Сугено е представен на Фиг. 1.



Фиг.1. Основно представяне на МПУ

Невронно-размитият модел на Такаги-Сугено е изключително подходящ за моделиране на нелинейни системи. От друга страна е добре известно, че широк

клас от нелинейни системи се представят с дискретната форма на описанието в пространството на състоянията. В настоящата разработка е използван именно такъв модел в пространството на състоянията за построяването на предсказващия модел на Такаги-Сугено.

Обобщеният запис за дискретната форма на описанието в пространството на състоянията има следния вид:

$$\begin{aligned} x(k+1) &= f_x(x(k), u(k)) \\ y(k) &= f_y(x(k), u(k)), \end{aligned} \quad (1)$$

където $x(k)$, $u(k)$ и $y(k)$ са векторът на състоянието, входовете и изходите на обекта със съответните им размерности $\dim(x_k)=n \times 1$, $\dim(u_k)=m \times 1$, $\dim(y_k)=q \times 1$. Неизвестните нелинейни функции f_x и f_y се апроксимират чрез следното множество от размити правила:

$$\begin{aligned} R_i : \text{if } z_1(k) \text{ is } M_{i1} \text{ and } \dots z_p(k) \text{ is } M_{ip} \\ \text{then } \begin{cases} x_i(k+1) = A_i x(k) + B_i u(k) \\ y_i(k) = C_i x(k) + D_i u(k) \end{cases} \end{aligned} \quad (2)$$

където R_i е i -тото размито правило, z_p представя входно-изходните регресори, M_{ip} е функцията на принадлежност от съответно размито множество, а A_i , B_i , C_i и D_i са матриците от описанието в пространството на състоянията с $\dim(A_i)=n \times n$, $\dim(B_i)=n \times m$, $\dim(C_i)=q \times n$, $\dim(D_i)=q \times m$.

Състоянието на системата в следващ момент от време се изчислява като претеглена сума на активираните размити правила:

$$\begin{aligned} x(k+1) &= \sum_{i=1}^L \bar{\mu}_{yi}(k) (A_i x(k) + B_i u(k)) \\ \hat{y}(k) &= \sum_{i=1}^L \bar{\mu}_{yi}(k) (C_i x(k) + D_i u(k)). \end{aligned} \quad (3)$$

Отделните матрици от описанието също могат да се изчислят като претеглена сума на локалните матрици A_i , B_i , C_i и D_i от съответните активирани правила i :

$$\begin{aligned} A(k) &= \sum_{i=1}^L A_i \bar{\mu}_{yi}(k) & B(k) &= \sum_{i=1}^L B_i \bar{\mu}_{yi}(k) \\ C(k) &= \sum_{i=1}^L C_i \bar{\mu}_{yi}(k) & D(k) &= \sum_{i=1}^L D_i \bar{\mu}_{yi}(k) \end{aligned} \quad (4)$$

където $\bar{\mu}_{yi} = \frac{\mu_{yi}}{\sum_{i=1}^L \mu_{yi}}$ е нормираната степен на

принадлежност в i -то активирано правило, а L е броя на активираните от базата правила. Импликацията е представена чрез следния израз:

$$\mu_{yi} = \mu_{i11} * \mu_{i12} * \mu_{i1j} * \dots * \mu_{ipj} \quad (5)$$

където μ_{ipj} е степента на принадлежност за активираното j -то размито множество за p -ти входен сигнал съгласно избраната тук Гаусова функция на принадлежност:

$$\mu_{pj}(z_p) = \exp\left(-\frac{(z_p - c_{Gpj})^2}{2\sigma_{pj}^2}\right) \quad (6)$$

За (6) c_{Gpj} отразява центъра на функцията, а σ_{pj} е разпределението за j -тата функция на принадлежност за p -ти вход z_p .

В невронно-размития модел на Такаги-Сугено е необходимо да се определят неизвестните параметри – брой на функциите на принадлежност, тяхната форма и параметри, както и коефициентите във функциите f_x и f_y в следствената част на правилата. Редица методи са посветени на тази задача за идентификация.

В представената разработка е приложен сравнително опростен подход, чиято алгоритмична реализация и рекурентна форма, го правят подходящ за приложението в реално време. Той се базира на петслойна невронно-размита мрежа, в която всеки слой изпълнява специфична за размитата логика задача. В първия слой са разположени входните възли Z_1, Z_2, \dots, Z_p . Те насочват входовете на системата към втория слой на мрежата, където се извършва операцията размиване. Третият слой интерпретира базата с логически правила. Предпоследният слой е слой за деразмиване, а последният пети е извеждащ (изходен) слой за мрежата. Параметрите за настройка са разположени във втория и четвъртия слой на структурата.

Алгоритъм за обучение на предложени невронно-размит модел. За обучение на предложеният вътрешен невронно-размит модел е приложена двустъпкова градиентна оптимизационна процедура. Алгоритъмът се базира на минимизирането на текущата грешка, изчислявана като разликата между стойностите на

реалния изход на обекта $y(k)$ и моделирания изход $\hat{y}(k)$.

$$\varepsilon(k) = y(k) - \hat{y}(k) \quad (7)$$

$$E_{FNN}(k) = \varepsilon^2(k) / 2 \quad (8)$$

За целта е необходимо да бъдат настроени две групи параметри от невронно – размитата архитектура. Това са параметрите в предпоставъчната част на правилата и параметрите в тяхната следствена част. Параметри в следствената част са коефициентите попълващи матриците на състоянието във функциите f_x и f_y и се изчисляват със следното обобщено рекурентно уравнение [6]:

$$\beta_i(k+1) = \beta_i(k) + \eta \left(-\frac{\partial E_{FNN}}{\partial \beta_i} \right) \quad (9)$$

където η е скоростта на обучение, а β_i е оценявания i -ти параметър a_{i1} , b_{i1} , c_{i1} или d_{i1} .

$$a_{li}(k+1) = a_{li}(k) + \eta \varepsilon(k) \bar{\mu}_{yi}(k) z_p(k) \quad l = 1 \div n \times n \quad (10)$$

$$b_{li}(k+1) = b_{li}(k) + \eta \varepsilon(k) \bar{\mu}_{yi}(k) z_p(k) \quad l = 1 \div n \times m$$

$$c_{li}(k+1) = c_{li}(k) + \eta \varepsilon(k) \bar{\mu}_{yi}(k) z_p(k) \quad l = 1 \div q \times n$$

$$d_{li}(k+1) = d_{li}(k) + \eta \varepsilon(k) \bar{\mu}_{yi}(k) z_p(k) \quad l = 1 \div q \times m$$

Параметри в предпоставъчната част на правилата са центърът c_{Gpj} и ширината σ_{pj} на активираното Гаусово размито множество за съответния вход на невронно-размитата структура. Те са обединени в параметъра α_{pj} .

$$\alpha_{pj}(k+1) = \alpha_{pj}(k) + \eta \left(-\frac{\partial E_{FNN}}{\partial \alpha_{pj}} \right) \quad (11)$$

където E_{FNN} е грешката от изхода на мрежата, която е разпространена обратно към втория слой от структурата.

$$\frac{\partial E_{FNN}}{\partial \alpha_{pj}} = \frac{\partial E_{FNN}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \mu_{pj}} \cdot \frac{\partial \mu_{pj}}{\partial \alpha_{pj}} \quad (12)$$

Съгласно уравнения (11-12) актуализацията за параметрите от предпоставъчната част придобива вида

$$c_{Gpj}(k+1) = c_{Gpj}(k) + \eta \varepsilon(k) \bar{\mu}_{yi}(k) [f_{yi} - \bar{y}(k)] \frac{[z_p(k) - c_{Gpj}(k)]}{\sigma_{pj}^2(k)} \quad (13)$$

$$\sigma_{pj}(k+1) = \sigma_{pj}(k) + \eta \varepsilon(k) \bar{\mu}_{yi}(k) [f_{yi} - \bar{y}(k)] \frac{[z_p(k) - c_{Gpj}(k)]^2}{\sigma_{pj}^3(k)} \quad (14)$$

ОПТИМИЗАЦИОНЕН АЛГОРИТЪМ

Чрез получения невронно-размит модел на Такаги-Сугено оптимизационният алгоритъм изчислява бъдещите стойности на управляващото въздействие във всеки такт на дискретизация. За целта се минимизира широко прилагания в МПУ целеви критерий от квадратичен вид със следната форма:

$$J(k) = \sum_{i=Hw}^{H_p+H_w-1} \|\hat{y}(k+i) - r(k+i)\|^2 Q + \sum_{i=0}^{H_u-1} \|\Delta \hat{u}(k+i)\|^2 R \quad (15)$$

където $\hat{y}(k)$, $r(k)$ и $\Delta \hat{u}(k)$ са съответно векторите на предсказания изход на модела, заданието в системата за управление и предсказаното управляващо въздействие в момента k . $Q \geq 0$ и $R > 0$ са константни тегловни матрици, които осъществяват връзката между изискванията за точност на управление и големини на управляващите въздействия. Хоризонтът на предсказване, началният хоризонт на предсказване и хоризонтът на управление са отбелязани съответно с H_p , H_w и H_u .

Целевата функция (15) може да се презапише в следния вид:

$$J(k) = \|Y(k) - T(k)\|^2 Q + \|\Delta U(k)\|^2 R \quad (16)$$

където

$$Y(k) = \begin{bmatrix} \hat{y}(k) \\ \vdots \\ \hat{y}(k+H_p-1) \end{bmatrix} \quad T(k) = \begin{bmatrix} r(k) \\ \vdots \\ r(k+H_p-1) \end{bmatrix} \quad \Delta U(k) = \begin{bmatrix} \Delta \hat{u}(k) \\ \vdots \\ \Delta \hat{u}(k+H_u-1) \end{bmatrix}$$

$$Q = \begin{bmatrix} Q(1) & 0 & \dots & 0 \\ 0 & Q(2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q(H_p) \end{bmatrix} \quad R = \begin{bmatrix} R(1) & 0 & \dots & 0 \\ 0 & R(2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & R(H_u) \end{bmatrix}$$

Предсказаният изход също може да се запише в следната матрична форма

$$Y(k) = \Psi x(k) + \Gamma u(k-1) + \Theta \Delta U(k) \quad (17)$$

където

$$\Psi = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{H_p-1} \end{bmatrix} \quad \Gamma = \begin{bmatrix} D \\ CB+D \\ CAB+CB+D \\ \vdots \\ C \sum_{i=0}^{H_p-2} A^i B + D \end{bmatrix} \quad \Theta = \begin{bmatrix} D & 0 & \dots & 0 \\ CB+D & D & \dots & \vdots \\ CAB+CB+D & \vdots & \ddots & 0 \\ \vdots & \vdots & \ddots & D \\ C \sum_{i=0}^{H_u-2} A^i B + D & \dots & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C \sum_{i=0}^{H_p-2} A^i B + D & \dots & C \sum_{i=0}^{H_p-1} A^i B + D \end{bmatrix}$$

За предсказаната грешка $E(k)$ е валиден записът:

$$E(k) = T(k) - \Psi x(k) - \Gamma u(k-1) \quad (18)$$

Друга формулировка на целевия критерий се получава при заместване на изрази (17) и (18) в (15):

$$J(k) = \Delta U^T H \Delta U - \Delta U^T \Phi + E^T Q E \quad (19)$$

където

$$\Phi = 2\Theta^T Q E(k) \quad H = \Theta^T Q \Theta + R$$

Задачата за минимизирането на така получената целева функция (19) се разглежда като задача от квадратичното програмиране. Ако Хесианът H е положително дефиниран, то задачата за оптимизацията се разглежда като от „изпъкнал“ („convex“) тип, и решението ѝ може да се запише в следната съкратена форма:

$$\Delta U = \frac{1}{2} H^{-1} \Phi \quad (20)$$

Най-общо ограниченията в задачата за квадратичното програмиране (QP), може да се представят в следния вид:

$$W \Delta U(k) \leq w \quad F U(k) \leq f \quad G Y(k) \leq g \quad (21)$$

Размерността на матриците W и F може да се различава от тази на G , въпреки еднаквата им структура. Тъй като $U(k)$ и $Y(k)$ не участват в явен вид в оптимизационната задача, дефинираните по-горе ограничения могат да се запишат относно $\Delta U(k)$. Това води до следния запис:

$$\begin{bmatrix} F_1 \\ G \Theta_c \\ W \end{bmatrix} \Delta U \leq \begin{bmatrix} -F_2 u(k-1) + f \\ -G(\Psi_c x(k) + \Gamma_c u(k-1)) + g \\ w \end{bmatrix} \quad (22)$$

Взимайки в предвид въведените ограничения (22), целевият критерий (15) придобива следния вид:

$$\begin{aligned} \min J(k) &= \Delta U^T H \Delta U - \Delta U^T \Phi + E^T Q E \\ \text{subject to } \Omega \Delta U &\leq \omega \end{aligned} \quad (23)$$

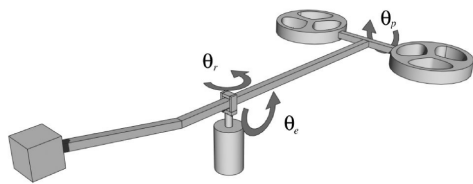
където $\Omega \Delta U \leq \omega$ отразява (22).

Така дефинирана оптимизационната задача е задача от квадратичното програмиране с ограничения от тип линейни неравенства (LICQP).

СИМУЛАЦИОННИ РЕЗУЛТАТИ

Разглежданият лабораторен стенд – двуроторна аеродинамична система с две степени на свобода (Фиг.2.). Тя се състои две перки, които осигуряват маневрирането на модела. Срещуположно на тях по рамото е разположена противотегжест. Моделът може да се върти около оста на неподвижната основа. На Фиг. 2. с θ_e е отбелязан ъгълът между рамото и хоризонталната равнина, θ_r е ъгълът на завъртане около оста, а θ_p е ъгълът на гредата свързваща двете перки [7]. Динамиката на модела може да се опише със следните уравнения:

$$\begin{aligned} \ddot{\theta}_e &= (K_f I_a / J_e)(V_f + V_b) - T_g / J_e \\ \ddot{\theta}_r &= -(F_a I_a / J_r) \sin \theta_p \\ \ddot{\theta}_p &= (K_f I_h / J_p)(V_f - V_b) \end{aligned} \quad (24)$$



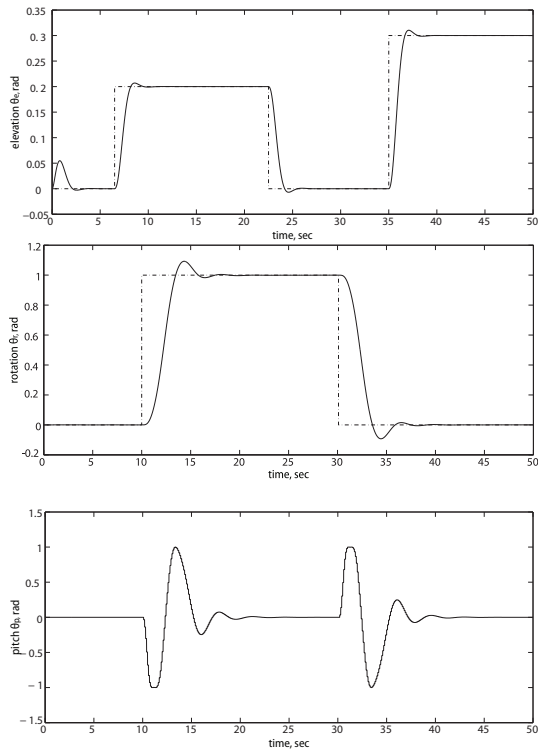
Фиг.2. Схема на изследвания обект – двуроторна аеродинамична система

Входове за разглежданата система са напреженията на двата мотора V_f и V_b , съответно за двете перки. Напрежението варира в интервала $[-2V; +3V]$. Всички шест състояния $x = [\theta_e \ \dot{\theta}_e \ \theta_r \ \dot{\theta}_r \ \theta_p \ \dot{\theta}_p]$ на системата са измерими. Ограничения са приложени на ъглите θ_e и θ_p .

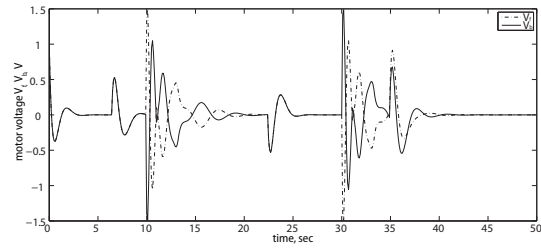
$$-0.5 \text{ rad} \leq \theta_e \leq 0.6 \text{ rad} \quad -1 \text{ rad} \leq \theta_p \leq 1 \text{ rad} \quad (25)$$

Симулационните резултати са получени при експерименти проведени със следните параметри: $H_w=1$, $H_p=30$, $H_r=10$, $T=50s$. За моделиране на нелинейния обект е използван 5-слойна Такаги-Сугено размита невронна мрежа с по три Гаусови функции за принадлежност за всеки входно-изходен регресор.

Полученият невронно-размит модел осигурява матриците на състоянието и управлението за описанието в пространството на състоянията, които се използват при решаване на оптимизационната задача (23). Тегловните матрици, участващи в целевия критерий, са избрани както следва: $Q(i)=\text{diag}([1,1])$ and $R(i)=\text{diag}([0.1,0.1])$. При експеримента се управляват положениата θ_e и θ_r , при въведените горе ограничения за системата.



Фиг.3. Изходи на изследваната система при променящо се задание



Фиг. 4. Входове на изследваната система при променящо се задание

Получените резултати показват работоспособността на предложения подход за НМПУ.

Последната графика на Фиг. 3. показва изхода на неуправлявания ъгъл θ_p , за който обаче е в сила ограничение ($-1 \text{ rad} \leq \theta_p \leq 1 \text{ rad}$). Вижда се, че системата достига задание, без да се нарушават ограниченията както за състоянията, така и за входовете на системата.

ЗАКЛЮЧЕНИЕ

Невронно-размитият подход за моделиране на нелинейни обекти дава възможност за работа в реално време и води до добри резултати. Използването на модел на Такаги-Сугено в пространството на състоянията, опростовява в значителна степен оптимизационната процедура при НМПУ, осигурявайки адекватен модел на всяка стъпка.

Моделът в пространството на състоянията позволява разглеждането на оптимизационната задача от МПУ като задача от квадратичното програмиране и най-важното - позволява включването на ограничения спрямо системните променливи. Това е от голямо значение, тъй като всички реални процеси имат ограничения спрямо параметрите си.

БЛАГОДАРНОСТ

Авторите изказват своята благодарност на НИС при Технически университет София, за осигуреното финансиране по научен проект No: 102ни046-08.

ЛИТЕРАТУРА

1. Maciejowski J. M., Predictive Control with Constraints, Prentice Hall Inc., New York, USA, 2001.
2. Martinsen F., Lorenz T. Biegler, Bjarne A. Foss, A new optimization algorithm with application to nonlinear MPC, Journal of Process Control, vol.14, pp 853–865, 2004.
3. Mendonça L.F., J.M. Sousa J.M.G. Sá da Costa, Optimization Problems in Multivariable Fuzzy Predictive Control, Int. Journal of Approximate Reasoning, vol.36, pp. 199–221, 2004.
4. Qin S. J. and T. A. Badgwell. A survey of industrial model predictive control technology. Control Eng. Prac., 11(7):733–764, 2003.
5. Fletcher R. Practical Methods of Optimization. 2nd ed., Wiley, 2000
6. Ahmed S., M. Petrov, A. Ichtev, "Model predictive control of a laboratory model – coupled water tanks," in *Proceedings of International Conference Automatics and Informatics '09*, October 1–4, 2009, Sofia, Bulgaria. pp. VI-33 - VI-35.
7. Johan Åkesson, MPCtools 1.0 – Reference Manual, ISSN 0280-5316