

Mély Neurális Háló Modul Generátor

Deep Neural Network IP Generator

*Dr. habil. VÁSÁRHELYI József*¹, *REDA Ahmad*², *Dr. BOUZID Ahmed*³, *DRÓTOS Daniel*⁴

^{1,2,3,4}Miskolci Egyetem, Automatizálási és Infó-kommunikációs Intézet
3515 Miskolc, Egyetemváros
{vajo, autareda, qgebouzid, iitdani}@uni-miskolc.hu
<https://mazsola.iit.uni-miskolc.hu>

Abstract

Model predictive control (MPC) is a classic control strategy used to solve online optimization problems. The paper provides information about a developed deep neural network (DNN) tool - based control strategy for automated steering deployed on FPGA. The DNN model was designed and trained based on the behavior of the traditional MPC controller. The DNN model was implemented on FPGA using different data types. The obtained results show that the suggested DNN model provided a satisfactory performance and successfully imitated the behavior of the traditional MPC.

Keywords: deep neural network, Field Programmable Gate Array, autonomous vehicle, model predictive control

Kivonat

A modell prediktív szabályozás (MPC) klasszikus vezérlési stratégia, amelyet online optimalizálási problémák megoldására használnak. A cikk ismerteti egy olyan eszközt, amelynek segítségével neurális hálókat generálhatók (Deep Neural Network -- DNN), a generált DNN, MPC szabályozási irányítás FPGA-n történő megvalósításra használható. A DNN modellt a hagyományos MPC szabályzó viselkedése alapján jelen cikk szerzői fejlesztették ki. A DNN modellt FPGA-n valósították meg különböző adattípusok felhasználásával. A kapott eredmények azt mutatják, hogy a javasolt DNN modell kielégítő teljesítményt nyújtott, és sikeresen utánozta a hagyományos MPC viselkedését.

Kulcsszavak: mély neurális háló, FPGA – logikai kapumátrix, autonóm járművek, modell prediktív szabályozás

1. BEVEZETÉS

A modell prediktív irányítást (MPC) online optimalizálási problémák megoldására használnak. A cikk információkat nyújt egy autonóm járművek kormányzásához kifejlesztett mély neurális hálózat (DNN) eszközalapú szabályozási stratégia megvalósításáról FPGA-n. A DNN modellt a hagyományos MPC szabályzó viselkedése alapján terveztük és tanítottuk be. Az FPGA hardveren történő megvalósításhoz egy új szellemi tulajdon IP generátor alapú eszközt fejlesztettek ki a cikk szerzői. A megvalósítás teljesítmény értékelését annak alapján végeztük, hogy a szabályzók képesek-e a jármű oldalirányú eltérését és elfordulási szögét a lehető legközelebb a nullához vinni. A DNN modellt FPGA-n valósítottuk meg különböző adattípusok felhasználásával (két különböző megvalósítás – fix pontos és lebegőpontos). A kapott eredmények azt mutatják, hogy a javasolt DNN modell kielégítő teljesítményt nyújtott, és sikeresen emulálja a hagyományos MPC viselkedését.

Az autonóm jármű alapvető funkciói három fő kategóriába sorolhatók: észlelés, tervezés és irányítás. A környezetérzékelés biztosítja a jármű számára a szükséges információkat a jármű helyzetéről és környezetéről, stb. Különböző érzékelők és eszközök alkalmazhatók az észlelési feladat megoldására, például ultrahangos érzékelők, kamerák használatával vagy az érzékelők fúziójával komplex helyzetértékelés valósítható meg [1]. Az összegyűjtött adatok alapján a legoptimálisabb döntés születik, és a tervezési modulban megtörténik a szükséges ellenőrzési műveletek elvégzése annak érdekében, hogy a jármű hatékonyan a kívánt pályát válassza

ki. A vezérlési funkcióban a parancsok elküldésre kerülnek a végrehajtó elemekhez a szabályozási stratégia megvalósításához [2].

A modell prediktív irányítás (MPC) az egyik leggyakrabban használt irányítási stratégia, mivel képes megoldani egy online optimalizálási problémákat. A nemlineáris rendszerek esetében azonban az MPC megvalósítása nagy kihívást jelent, gyakran kivitelezhetetlen a magas számítási igények miatt, különösen az erőforrásukban korlátozott beágyazott rendszerekben [3]. Következésképpen az MPC megvalósításánál cél a hatékony és alacsony disszipált teljesítményű struktúrák kidolgozása a számítási terhelések és a végrehajtási idő minimalizálása szempontjából. A gépi tanulási módszerek jelentős teljesítménye sokrétű alkalmazással a különböző területeken indokolta az autóiipari vezérlési rendszerekben történő alkalmazásukat. A mély neurális hálózatok környezetérzékeléssel és mozgástervezéssel kapcsolatos pozitív eredmények felhívták a figyelmet a DNN alapú szabályozási stratégiák megvalósítása iránt [4], [5]. A mély neurális hálózatok számos egyéb előnyt kínálnak a végrehajtási idő és a számítási terhelés csökkentése terén, ami hatékonyabbá teszi a korlátozott erőforrású HW-n (Hardveren) való megvalósítást [6]. A saját DNN alapú IP- megvalósítás kulcsfontosságú az FPGA alapú implementációhoz.

Egyes tanulmányok modell prediktív szabályzókat FPGA-kon történő megvalósítását javasolják, figyelembe véve a különböző megvalósítási módszereket, mint például a magas szintű szintézis [6] a Xilinx rendszergenerátor [7] vagy akár a HDL [8]. Ebben a cikkben a javasolt megoldás egy mély neurális hálózati modell kidolgozása a hagyományos MPC szabályzó viselkedésén alapulva, hogy a DNN modell kiválthassa az MPC szabályzót nagy bonyolultságú irányítási rendszerben. Amennyiben szekvenciális számítási rendszerekben valósítják meg a DNN hálózatokat úgy számítási idő és erőforrás igényük megnő.

Jelen cikk felépítése a következő: A Bevezetés után, a második fejezet bemutatja az MPC szabályzó tervezését, míg a harmadik fejezet a DNN IP generátort mutatja be, míg a 4. fejezet az eredményeket és az utolsó fejezet, pedig a következtetéseket tárgyalja.

2. MPC SZABÁLYZÓ TERVEZÉSE

A modell-prediktív szabályzó az autonóm járművet a kívánt pálya mentén a célpontra irányítja az oldalirányú eltérés d és a jármű relatív elfordulási szögének θ szabályozásával. Ezeknek a változóknak a nullán vagy a nullához a lehető legközelebbi értéken kell maradniuk, ami valós idejű optimalizálási probléma, amelyet az MPC szabályzónak kell kezelnie. Mivel az MPC egy modellalapú szabályzó, a tervezési folyamat két fő lépésből áll, először az rendszer modellt (a járművet) kell megtervezni, majd a második lépésben az MPC szabályzót. A tervezési folyamat magában foglalja a szabályzó paramétereinek hangolását és a rendszer által meghatározott működési feltételek megfogalmazását enyhe és szigorú feltételek megadásával.

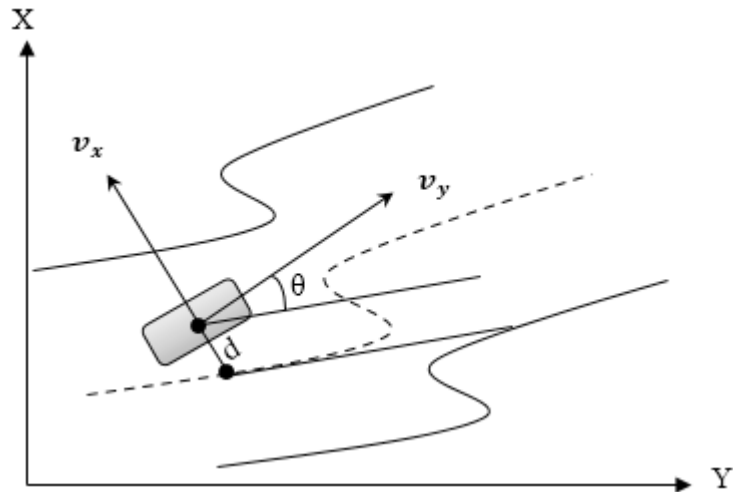
2.1 A jármű modell bemutatása

A jármű dinamikus modelljét az az alábbi egyenletek mutatják be (1, 2, 3). Az 1. ábra a jármű globális helyzetét mutatja, ahol, v_x és v_y a hosszirányú és oldalirányú sebességek, d az oldalirányú eltérés, m a jármű össztömege, l_r és l_f rendre a hátsó gumiabroncs és az első gumiabroncs és a tömegközéppont közötti távolság, I_z lengési nyomaték, c_f és c_r az első és hátsó gumiabroncsok sarokmerevsége, δ a kormányzási szög, θ az elfordulási szög, ρ a görbület és ω a lengési sebesség.

$$\frac{d}{dt} \begin{bmatrix} v_y \\ \omega \\ d \\ \theta \end{bmatrix} = [A] \begin{bmatrix} v_y \\ \omega \\ d \\ \theta \end{bmatrix} + [B] \begin{bmatrix} \delta \\ \rho v_x \end{bmatrix} \quad (1)$$

$$A = \begin{bmatrix} \frac{-2c_f + 2c_r}{mv_x} & -v_x - \frac{2c_f l_f - 2c_r l_r}{mv_x} & 0 & 0 \\ -2c_f l_f - 2c_r l_r & -2c_f l_f^2 - 2c_r l_r^2 & 0 & 0 \\ \frac{I_z v_x}{1} & \frac{I_z v_x}{0} & 0 & v_x \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (2)$$

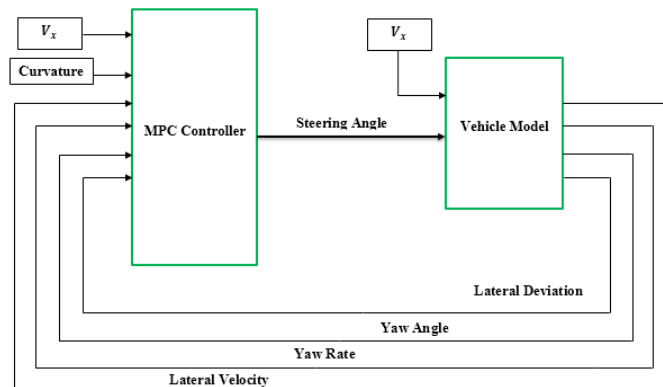
$$B = \begin{bmatrix} \frac{2c_f}{m} & 0 \\ 2c_f l_f & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad (3)$$



1. ábra A járműmodell helyzete.

2.2 Az MPC modell

Az MPC modell tervezésének első lépése a járműmodell bemeneti-kimeneti jeleinek meghatározása. A második lépés pedig a paraméterek beállítása és a korlátok meghatározása. A manipulált változó (δ) és a zaj ($v_x \rho$) bemeneti jelként, míg az oldalsebesség v_y , az oldalirányú eltérés d , az irányváltozás szöge ω és az elfordulási sebességet kimeneti jelként határozzuk meg. Az MPC modell paramétereit időrendben határoztuk meg: a mintavételezési idő $T_s = 0,1$ s, az idő horizont $P = 2$ s, és a kontroll horizont $M = 2$ s. A korlátokat a következők: kormányaszög a $[-1,04, 1,04]$ rad intervallumban vehet fel értékeket. Az elfordulási szög sebességet pedig a $[-0,26, 0,26]$ rad intervallumra korlátoztuk. A paramétereket a tervezési folyamat során a kielégítő viselkedés eléréséig fenntartottuk. Az MPC és az üzemmódel átfogó kialakítása a 2. ábrán látható.

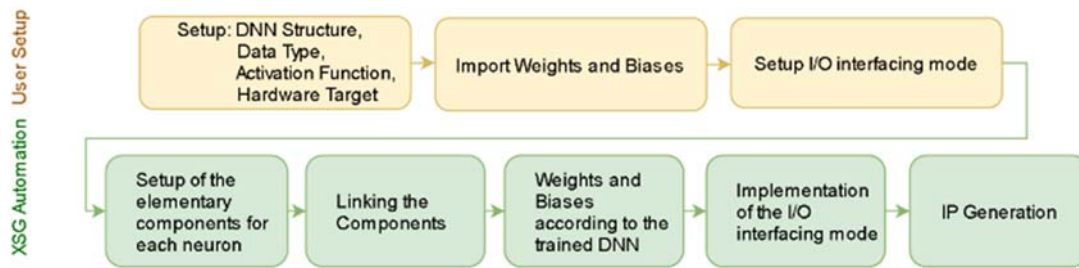


2. ábra Az MPC általános blokkvázlata és az rendszermodell tömbvázlata

3. DNN IP MODUL GENERÁTOR

A neurális hálózat IP automatikus generálásának folyamata két fő lépésből áll (3. ábra). Először a felhasználónak meg kell határoznia a DNN-re vonatkozó paramétereket (a szerkezetet, az adattípust és az aktiválási függvényt), valamint a megcélzott számítási HW-t, a súlyok és a torzítási paraméterek értékeit importáljuk az előre betanított DNN-ből. Ezután jön a bemeneti/kimeneti interfész mód beállításának lépése, ahol 4 mód áll rendelkezésre: UART, AXI (Advanced eXtensible Interface), korlátozott párhuzamos és interfész nélküli módok. Ha az UART interfész módot választjuk, akkor két további előre megtervezett IP kerül meghívásra, amelyek felelősek az UART adatok fogadásáért és továbbításáért a DNN IP-ről vagy küldéséért. Ebben a módban a felhasználónak meg kell adnia a Tx és Rx számára használandó portokat. Ha az AXI módot választottuk ki, ez lehetőséget biztosít a DNN IP-nek a feldolgozó rendszerrel vagy egy mikroprocesszor maghoz történő illesztésre.

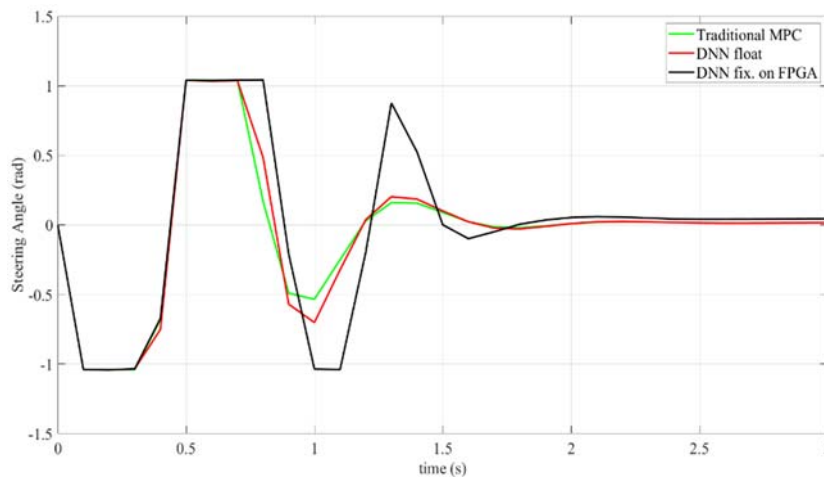
A DNN-IP paraméterek beállítása után kezdődik az Xilinx System Generátor elemekből történő automatikus IP generálás, amely az egyes neuronokhoz szükséges elemi számítási komponensek meghívásából, a komponensek és a neuronok összekapcsolásából, a súlyok és torzítások ennek megfelelő beállításából, a megfelelő I/O megvalósításából áll, majd létrehozza az IP-t.



3. ábra A DNN tervezés lépéseinek ábrája

4. EREDMÉNYEK

A hagyományos MPC implementációk, a DNN modell és a DNN FPGA-n történő megvalósítását a teljesítmény szempontjából tárgyaljuk. Referenciának a hagyományos MPC választ tekintjük. A teljesítmény mellett a lebegőpontos és fixpontos adattípust használó telepítéseket is vizsgáltuk a felhasznált erőforrások szempontjából. A szabályozók teljesítményét a T_s beállási idő, az M_p túllövés, valamint a jármű elfordulási szögének és oldalirányú eltérésének végső értéke (stacionárius) alapján értékeltük. A 4. ábrán jól látható, hogy a DNN modell és a hagyományos MPC nagyon hasonló válaszreakciót mutat, ami azt jelenti, hogy a DNN modell sikeresen utánozta a hagyományos MPC viselkedését, míg az FPGA-n lévő DNN változat némileg eltérő reakciót mutat.



4. ábra Az FPGA-n számított becsült kormány elfordulási szög változásának grafikonja a hagyományos MPC és a DNN modell szimulációs eredményeihez képest.

Az 1. Táblázat a fixpontos és lebegőpontos DNN erőforrás felhasználást mutatja AXI interfész illesztés esetében (LUT = keresőtábla; DSP = jelfeldolgozó egységek)

FPGA erőforrás felhasználás a DNN fixponthoz és lebegőpontos megvalósítások esetében

1. táblázat

Site Type	Felhasznált Erőforrások		FPGA Erőforrás	Erőforrás felhasználás [%]	
	Fix Pontos	Lebegő Pontos		Fix Pontos	Lebegő Pontos
LUTs	7166	183034	203800	3.52	89.81
DSPs	393	826	840	46.79	98.33

A táblázatban mutatott értékek alapján elmondható, hogy a lebegő pontos megvalósítások kimerítik a rendelkezésre álló erőforrásokat és egyéb modulok illesztése már nem lehetséges, míg a fix pontos megvalósítás már lehetőséget biztosít például érzékelők illesztésére is.

5. KÖVETKEZTETÉSEK

Ebben a cikkben egy mély neurális hálózatot automatikus IP generálását vizsgáltuk. A DNN IP viselkedését összehasonlítva a hagyományos MPC szabályzó viselkedésével elmondható hogy minél jobb a

DNN generálás adatstruktúrája (lebegő pontos) annál jobban közelít a hagyományos MPC szabályozásához. Míg ha a fixpontos működést vizsgáljuk, azt látjuk, hogy a kezdeti indulásnál a szabályzó túllő.

A Xilinx XSG-n alapuló új eszközt fejlesztettünk ki a DNN modell FPGA-kon történő megvalósításra. Az eredmények azt mutatják, hogy a betanított modell nagyon kicsi négyzetes hibával ($RMSE = 0,011228$ rad) sikeresen teljesítette az MPC szabályzó működését. A betanított DNN modellt valósítottuk meg Xilinx Kintex-7 FPGA KC705 kártyán fixpontos adattípus használatával, Az elért eredmények kielégítőek és megfelelnek a tervezési követelményeknek.

IRODALMI HIVATKOZÁSOK

- Best A, Narang S, Barber D, Manocha D. AutoVi: Autonomous vehicle planning with dynamic maneuvers and traffic constraints. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). doi: 10.1109/IROS.2017.8206087, 2017; 2629-2636.
- Best A, Narang S, Barber D, Manocha D. AutoVi: Autonomous vehicle planning with dynamic maneuvers and traffic constraints. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). doi: 10.1109/IROS.2017.8206087, 2017; 2629-2636..
- Swief A, El-Zawawi A, El-Habrouk M, and Eldin A. Approximate Neural Network Model for Adaptive Model Predictive Control. 6th Int. Comput. Eng. Conf. doi:10.1109/ICENCO49778.2020.9357373, 2020; 135-140.
- Swief A, El-Zawawi A, El-Habrouk M, and Eldin A. Approximate Neural Network Model for Adaptive Model Predictive Control. 6th Int. Comput. Eng. Conf. doi:10.1109/ICENCO49778.2020.9357373, 2020; 135-140,
- Lamouik I, Yahyaouy A, Sabri M. Deep neural network dynamic traffic routing system for vehicles. . doi:10.1109/ISACV.2018.8354012, 2018 International Conference on Intelligent Systems and Computer Vision (ISCV). 2018; 1-4.
- Bin Khalid A, Liegmann E, Karamanakos P, Kennel R. High-Level Synthesis of a Long Horizon Model Predictive Control Algorithm for an FPGA. PCIM Europe digital days 2020; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management. 2020; 1-8
- Singh V, Tripathi R, Hanamoto T. Implementation Strategy for Resource Optimization of FPGA-Based Adaptive Finite Control Set-MPC using XSG for a VSI System. IEEE J. Emerg. Sel. Topics in Power Electronics. 2021; 9(2): 2066-2078.
- Ingole D Holaza J, Takács B, Kvasnica M. FPGA-Based Explicit Model Predictive Control for Closed-Loop Control of Intravenous Anesthesia. 20th International Conference on Process Control (PC). doi: 10.1109/PC.2015.7169936. 2015; 42-47.