

Automatikus pályatervezés és mozgó horizontú prediktív irányítás járművek akadályelkerülő rendszere számára korszerű érzékelők bevonásával

Lantos Béla

Budapesti Műszaki és Gazdaságtudományi Egyetem
Irányítástechnika és Informatika Tanszék
1117 Budapest, Magyar Tudósok krt. 2
lantos@iit.bme.hu

Kivonat: – A cikk a pályatervezés és irányítás kérdéseivel foglalkozik földi járművek akadályelkerülő rendszere (CAS) esetén. Az automatikus pályatervezés az elasztikus szalag elvén alapul és módosított reakció erőket használ az útszegély és a statikus akadályok figyelembevételére az erőegyensúlyban. A pálya (referencia jel) simításra kerül és megvalósítását mozgó horizontú prediktív irányítás (RHC) végzi. A jármű teljes (nem-affin) vagy közelítő (bemenet affin) nemlineáris dinamikus modellel írható le. A nemlineáris prediktív irányítási probléma megoldása időben változó lineáris rendszer (LTV) alkalmazásával történik, melyet minden horizont kezdetén egy alkalmasan megválasztott nominális irányítási és állapot szekvencia körül történő linearizálással lehet meghatározni. A rendszer alulaktuált (a bemenő jelek száma kisebb a rendszer szabadságfokánál), ezért a referencia pályához a referencia irányítás nem vagy csak nehezen határozható meg, ami a linearizálást megnehezíti. Az optimalizálás a horizonton belül a hibára vonatkozó kvadrátikus kritérium és alkalmasan választott végfeltétel mellett analitikusan történik. A legelső horizont számára a nominális irányítás differenciálgeometriai megközelítésen alapuló módszerrel (DGA) kerül meghatározásra az irodalomból a bemenet affin modell számára ismert módszerrel. Állapotbecslés kétszintű Kalman-szűrőt, valamint kétantennás GPS és inercia navigációs rendszer (INS, gyorsulásérzékelő és giroszkóp) jeleire épül. A kifejlesztett algoritmusok alapján MATLAB licenstet nem igénylő stand-alone szoftver került kifejlesztésre, amely kielégíti a valós idejű elvárásokat.

Kulcsszavak: – Automatikus akadályelkerülés; Földi jármű; Lie-algebra; Mozgó horizontú irányítás; GPS/INS; Stand-alone programrendszer

1. Bevezetés

Az ütközéselkerülő rendszer (Collision Avoidance System, CAS) számára a pályatervezést automatikusan és valós időben kell megvalósítani statikus és dinamikus akadályok jelenlétében. A pálya információ ezután referencia jellé konvertálható az irányítások számára, amelyek differenciálgeometriai (Lie-algebrai), optimális prediktív irányítási stb. módszereken alapulhatnak. Statikus és dinamikus akadály lehet egy elől haladó gépkocsiról leeső teher és a másik sávban szembejövő jármű. A probléma teljes megoldása szenzorfüzión alkalmazását igényli radar, lézer és látórendszer bevonásával, a jelen cikk azonban ezekkel a kérdésekkel nem foglalkozik. A szenzor információ az elasztikus szalag inicializálásában játszik szerepet, amely a választott pályatervezési módszer alapja [1]. A pálya stabilitási tulajdonságai a karakterisztikus sebesség stabilitási indikátor (CVSI) segítségével ítéltelők meg [2]. Ha a pálya generálása nem fejezhető be egy T_{\max} időhatáron belül, vagy a pálya becsült CVSI tulajdonságai nem kielégítőek, akkor vészfékezés következik. Ellenkező esetben a megtervezett pályát valamilyen irányítási módszerrel meg kell valósítani.

Az irányítás klasszikus (PID stb.) módszeren, vagy korszerű Lie-algebrai, optimális irányítási vagy prediktív irányítási módszeren alapulhat. A jármű egyszerűsített nemlineáris bemenet affin modellje esetén a nemlineáris szétcsatolási probléma analitikusan megoldható [4]. Állapotbecslésre kétszintű Kalman-szűrő, kétantennás GPS és inercia navigációs rendszer (Inertial Navigation System, INS) alkalmazása egy tipikus megközelítés [3].

Egy lényeges kérdés az irányítások gyors prototípus tervezése, amelyre a MATLAB, Optimization Toolbox, Simulink, Real Time Workshop, target fordítók, valamint a Matlab Compiler adnak jó lehetőséget.

2. Automatikus ütközésselkerülő pályatervezés

A megvalósított pályatervezési algoritmus az elasztikus szalag elvén alapul. A módszer lehetővé teszi a pályatervezés során statikus és dinamikus akadályok kikerülését. A pályatervezés automatikus és valós időben zajlik. A statikus és dinamikus akadályok paraméterei az algoritmus bemeneti adatai. Ezek az információk rendszerint valós idejű képfeldolgozás eredményei, amellyel a rendszer a jövőben bővíthető. A pályatervezési algoritmus a következő lépésekre bontható:

1. Induló elasztikus szalag generálása és iterálása az erőegyensúly eléréséig.
2. Pályatervezés az egyensúlyi akadályelkerülő elasztikus szalaghoz.
3. A pályához tartozó állapotváltozók kiszámítása.

A kifejlesztett program a gyakorlat számára fontos szituációra, egy statikus akadály és egy szembejövő jármű esetére koncentrál. Ez a lépés a bemeneti adatként szolgáló specifikáció (statikus akadály elhelyezkedése és mérete az útpálya sávjaiban, szembejövő dinamikus akadály kiindulási helyzete, mérete és sebessége) ismeretében felvesz egy induló elasztikus szalagot. A bemeneti specifikációban a pályatervezés kezdeti t_0 pillanatában az akadályelkerülést végző járműhöz relatívan kell megadni a statikus és dinamikus akadályok geometriai adatait és a szembejövő jármű sebességét.

Az akadályelkerülést végző jármű $(0,0)^T$ kiindulási helyzetéhez (tipikusan a saját sáv közepéhez) relatívan kell meghatározni az akadályelkerülési pályát. Meghatározásra kerül a pályatervezésre rendelkezésre álló becsült T_{\max} maximális idő és az akadályelkerülést végző jármű becsült célhelyzete. A becsült célhelyzet és az akadályok helyzetének ismeretében tisztán geometriai szemlélet alapján történik az akadályelkerülés egy megengedett kiindulási pályájának kiválasztása és elasztikus szalaggal történő közelítése. Ennek során az akadályoktól és az úttest határaitól való távolságtartásra, valamint egyenesekből és körívекből felépülő pálya kiválasztására kell törekedni. A kiindulási pályát a rendszer iterációval tovább finomítja. Az elasztikus szalag szekcióinak darabszáma és a rugók kiindulási hossza programkonstans, nagyságrendjük kb. 40 csomópont illetve 1m.

2.1 Az akadályelkerülő elasztikus szalag generálása

Az elasztikus szalag (elastic band) egy rugókból láncszerűen összeillesztett rendszer. A rugók potenciál mezeje, az akadályok virtuális potenciál mezeje és az útszegélyek virtuális potenciál mezeje eredőjében a jármű pályája úgy lesz kiválasztva, hogy a potenciál mezőkből keletkező erőhatások egyensúlyban legyenek, és egyensúlyi helyzetben "esztétikus" (kis görbületű stb.) pálya keletkezzen. A pályatervezés egy iteratív folyamat, mivel véletlenszerűen statikus akadály keletkezhet és jármű bukkanhat fel. A környezetben bekövetkező változások beindítják a pályatervezési folyamatot.

A jármű (vehicle) pillanatnyi helyzetében az (x_v, y_v) koordináta-rendszer origójából (r_0) indul az elasztikus szalag. A szalag N darab rugóból áll, a rugók csatlakozási helyei a csomópontok (nodes), melyeket a jármű koordináta-rendszerében r_i azonosít az i -dik csomópont esetén. A rugószakasz rugóállandója k_i , a rugószakasz kezdeti hossza $l_{0,i}$, amelyek a pályatervezés számára megválaszthatók. Az erőegyensúlyi helyzet keresésekor minden iteratív lépésben meghatározásra kerül a csomópont t_i elérési ideje is, amely a csomópontok közötti lineáris interpoláción és a jármű megválasztható átlagos longitudinális sebességén alapul. Ha az erőegyensúlyi helyzethez tartozó pálya az iteráció során meghatározásra került, akkor a jármű végighalad az elasztikus szalag csomópontjain az átlagos longitudinális sebesség alapján számított időpontok figyelembevételével, a csomópontok között spline-technikával interpolálva.

Az elasztikus szalagot a következő információ jellemzi:

$$(r_0, t_0) \xrightarrow{k_0, l_{0,0}} (r_1, t_1) \xrightarrow{\dots} (r_i, t_i) \xrightarrow{k_i, l_{0,i}} (r_{i+1}, t_{i+1}) \xrightarrow{\dots} (r_N, t_N)$$

Az akadályokat biztonsági körök modellezik, az O_j akadály biztonsági körének középpontja r^{O_j} , átmérője pedig d_j , amelybe beleértendő az akadályt teljesen lefedő kör átmérője és a jármű szélessége is. A pályatervezés a rugók V_i^{int} belső potenciálján (az erők a megfelelő potenciálmezők iránymenti deriváltjai), a bal oldali (B_l) és a jobb oldali (B_r) sáv $F_i^{B_q}$ külső erőin, $q \in \{l, r\}$, és az

akadályok $F_{i,stat}^{O_j}$ és $F_{i,mov}^{O_j}$ külső erőin alapul. A potenciálfüggvény kiértékelésekor figyelembe kell venni, hogy az elasztikus szalag i csomópontja az r_i pontban a t_i időpontban lesz, ezért a mozgó akadálynak ehhez az időponthoz tartozó $r^{O_j}(t_i)$ akadályhelyzetét meg kell határozni. Ehhez szükség van az akadály mozgását jellemző kezdeti $r^{O_j}(t_0)$ akadályhelyzetre és $v^{O_j}(t_0)$ feltételezett akadály sebességre, amelyekből lineáris extrapolációval számítható ki $r^{O_j}(t_i)$. Az O_j akadály által az elasztikus szalag i csomópontjára ható erő a programban a mozgó akadály hatását az aktuális környezetére korlátozza. Minden r_i csomópont esetén az útszegélyek legközelebbi $r_i^{B_l}$ és $r_i^{B_r}$ pontjait választjuk. Mivel az [1] szerinti $F_i^{B_q}$ erők nem voltak képesek visszapozícionálni az elasztikus szalagot akadálymentes esetben a sáv közepére, valamint $F_{i,stat}^{O_j}$ nem biztosított kielégítő pályagörbületet és centripetális gyorsulást, ezért a következő választással élünk:

$$V_i^{int} = \frac{1}{2} k_i (|r_{i+1} - r_i| - l_{0i})^2 \quad (1)$$

$$V^{int} = \sum_{i=0}^{N-1} V_i^{int} = \sum_{i=0}^{N-1} \frac{1}{2} k_i (|r_{i+1} - r_i| - l_{0i})^2 \quad (2)$$

$$F_i^{B_q} = M^B \exp\left[-\frac{1}{2} (|r_i - r_i^{B_q}| / \sigma^{B_q})^2\right] \frac{r_i - r_i^{B_q}}{|r_i - r_i^{B_q}|} \quad (3)$$

$$\sigma^{B_q} = k^{B_q} / \sqrt{2 \ln(M^B / m^B)}, M^B = 2, m^B = 0.05$$

$$F_{i,stat}^{O_j} = k^{O_j} \frac{d^{O_j} / 2}{|r_i - r^{O_j}|} \cdot \frac{r_i - r^{O_j}}{|r_i - r^{O_j}|}, k^{O_j} = 3 \quad (4)$$

$$F_{i,mov}^{O_j} = k^{O_j} \exp\left[-\left(\frac{|r_i - r^{O_j}(t_i)| - \frac{d_j}{2}}{\frac{d_j}{2}}\right)^2\right] \frac{r_i - r^{O_j}(t_i)}{|r_i - r^{O_j}(t_i)|} \quad (5)$$

Legyen M a statikus és mozgó akadályok együttes száma, akkor az erő egyensúly feltétele

$$F_i^{sum} = F_i^{int} + F_i^{B_l} + F_i^{B_r} + \sum_{j=1}^M F_i^{O_j} = 0. \quad (6)$$

Az egyensúlynak minden i csomópontra szimultán kell fennállnia. Legyen ezért

$$x = (r_1^T, r_2^T, \dots, r_N^T)^T \text{ és } f(x) = ((F_1^{sum})^T, (F_2^{sum})^T, \dots, (F_N^{sum})^T)^T, \quad (7)$$

akkor a feladat az $f(x)=0$ nemlineáris egyenletrendszer megoldása, melyet a MATLAB Optimization Toolbox `fsolve` függvényével határoztunk meg, átadván számára nemcsak az $f(x)$ függvényt, hanem a számítások gyorsítása érdekében a függvény Jacobi-mátrixát (deriváltját) is. Ezáltal az erőegyensúly számítási ideje kb. 1s lett, amely a valós idejű alkalmazásokhoz megfelelő.

2.2 Referencia jel tervezése az irányításokhoz

Az elasztikus szalag egyensúlyi helyzete az akadályelkerülő pálya absztrakt formája, amelyet azonban az irányításokhoz referencia jel (alapjel) időfüggvényekké kell konvertálni. Célszerű ezen kívül még a mozgás további kinematikai jellemzőit (a pálya idő szerinti magasabb deriváltjait) is meghatározni, mivel ezek jelentősen támogatni tudják a rendszertechnikai vizsgálatokat.

A kifejlesztett programban az elasztikus szalag által generált pályához a saját jármű névleges sebességének figyelembevételével a MATLAB spline-szolgáltatásaival (`spline`, `ppval`, `unmkpp`, `mkpp`) meghatározzuk a pályához tartozó kinematikai jellemzőket és az időparaméter elosztását, melyek az irányítások számára referencia jelként (alapjel időfüggvényként) fognak szolgálni.

Az elasztikus szalag csomópontjait először poligonnal (törött vonallal) approximáljuk, amely egy (x,y) adatsorozat, majd a saját jármű átlagos sebességének felhasználásával meghatározzuk a

csomópontokhoz tartozó (t, x) , (t, y) adatsorozatokat. Ezután a MATLAB spline technikájának bevonásával a csomópontok között harmadfokú polinomokkal approximálunk, ami lehetővé teszi a deriváltak számítását a harmadik deriválttal bezárólag. Mivel azonban ez utóbbi már szakaszonként konstans, ami a dinamikus tulajdonságok miatt kedvezőtlen, ezért ennek simítására az első deriváltat ismét approximáljuk harmadfokú polinommal, ami sima harmadik deriváltat eredményez.

A deriváltak segítségével a kinematikai mennyiségeket (v sebességet, \dot{v} gyorsulást, κ görbületet, ψ orientációt és deriváltjait), a referencia mozgásban jogosan nulla β oldalcsúszási szöveget feltételezve, az alábbi összefüggések alapján számítjuk:

$$\begin{aligned} v &= (\dot{x}^2 + \dot{y}^2)^{1/2}; & \dot{v} &= \frac{\dot{x}\ddot{x} + \dot{y}\ddot{y}}{(\dot{x}^2 + \dot{y}^2)^{1/2}}; & \kappa &= \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{3/2}} \\ \psi &= \arctan(\dot{y}/\dot{x}); & \dot{\psi} &= \frac{\ddot{y}\dot{x} - \dot{y}\ddot{x}}{\dot{x}^2 + \dot{y}^2}; & \ddot{\psi} &= -2\frac{\dot{x}\ddot{x} + \dot{y}\ddot{y}}{(\dot{x}^2 + \dot{y}^2)^2} + \frac{\ddot{y}\dot{x} - \dot{y}\ddot{x}}{\dot{x}^2 + \dot{y}^2} \end{aligned} \quad (8)$$

3. Jármű dinamikus modellje és Lie-algebrai tulajdonságai

A jármű dinamikus modelljében az első kerekre (front wheels) F betűvel, a hátsó kerekre (rear wheels) R betűvel hivatkozunk. A nemlineáris dinamikus modell alapjait az [5] tanulmányban már összefoglaltuk, itt csak a programfejlesztés során alkalmazott módosításokat foglaljuk össze.

A jármű dinamikus modelljében, felhasználva [4] eredményeit, a β oldalcsúszási szög és a δ_w kormányzási szög trigonometrikus függvényeit elsőfokú Taylor-polinomokkal approximáljuk, de a többi változóban megőrizzük a nemlineáris összefüggéseket. Ezen kívül bemenő jelnek nem a longitudinális gyorsító erőt és a kormányzási szöveget tekintjük, hanem a longitudinális gyorsító erőt és az első kerékre ható oldalirányú származtatott erőt, amely utóbbiból a kormányzási szög számítható.

Vezessük be a következő további jelöléseket az első S_v és hátsó S_h oldalirányú erőkre:

$$S_h = c_R \left(-\beta - \frac{l_R \dot{\psi}}{v_G} \right), \quad S_v = c_F \left(\delta_w - \beta - \frac{l_F \dot{\psi}}{v_G} \right) \quad (9)$$

Nullának vesszük az első kereket meghajtó F_{IF} és nemnullának a hátsó kereket meghajtó F_{IR} erőt.

A rendszer bemenetének tekintjük az $u = (u_1, u_2)^T = (S_v, F_{IR})^T$ bemenő jel vektort, és állapotvektornak az $x = (\beta, \psi, \dot{\psi}, v_G, X, Y)^T$ vektort. Két nemlineáris modellt különböztetünk meg, az elsőben nem alkalmazzuk, a másodikban alkalmazzuk a Taylor-sorfejtést.

3.1 A pontos nemlineáris járműmodell

$$\begin{aligned} \dot{\beta} &= -\dot{\psi} + \frac{1}{m_v v_G} \{ F_{IF} \sin(\delta_w - \beta) - (F_{IR} - T) \sin(\beta) \\ &\quad + c_F \left(\delta_w - \beta - \frac{l_F \dot{\psi}}{v_G} \right) \cos(\delta_w - \beta) + c_R \left(-\beta + \frac{l_R \dot{\psi}}{v_G} \right) \cos(\beta) \} \\ \ddot{\psi} &= \frac{1}{I_{zz}} \{ l_F F_{IF} \sin(\delta_w) + l_F c_F \left(\delta_w - \beta - \frac{l_F \dot{\psi}}{v_G} \right) \cos(\delta_w) - l_R c_R \left(-\beta + \frac{l_R \dot{\psi}}{v_G} \right) \} \\ \dot{v}_G &= \frac{1}{m_v} \{ F_{IF} \cos(\delta_w - \beta) + (F_{IR} - T) \cos(\beta) - c_F \left(\delta_w - \beta - \frac{l_F \dot{\psi}}{v_G} \right) \sin(\delta_w - \beta) \\ &\quad + c_R \left(-\beta + \frac{l_R \dot{\psi}}{v_G} \right) \sin(\beta) \} \\ \dot{X} &= v_G \cos(\psi + \beta) \\ \dot{Y} &= v_G \sin(\psi + \beta) \end{aligned} \quad (10)$$

3.2 Approximált nemlineáris járműmodell

Az X, Y jelek kivételével Taylor-sorfejtéssel a következő approximált lineáris modellhez jutunk:

$$\dot{x} = \begin{pmatrix} -x_3 + (Tx_1 + S_h)/(m_v x_4) \\ x_3 \\ -S_h l_R / I_{zz} \\ -T / m_v \\ x_4 \cos(x_1 + x_2) \\ x_4 \sin(x_1 + x_2) \end{pmatrix} + \begin{pmatrix} 1/(m_v x_4) & -x_1/(m_v x_4) \\ 0 & 0 \\ l_F / I_{zz} & 0 \\ 0 & 1/m_v \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} S_v \\ F_{lR} \end{pmatrix} = A(x) + B(x)u \quad (11)$$

$$y = \begin{pmatrix} x_5 \\ x_6 \end{pmatrix} = C(x)$$

Mindkét modell nemlineáris, de az approximált nemlineáris modellnek az az előnye, hogy differenciálalgebrai (Lie-algebrai) értelemben létezik a MIMO nemlineáris relatív fokszáma, ezért kimeneti visszacsatolással linearizálható. A $T(x)$ légellenállást az irányítások során nem vesszük figyelembe. A vizsgálatok során mindkét modell esetén az $F_{lF} = 0$ és $T = 0$ választással élünk.

3.3 Az approximált modell Lie-algebrai tulajdonságai

Az approximált nemlineáris dinamikus modellre elvégezve az

$$N_A^k C_i(x) = \left[\frac{\partial}{\partial x} N_A^{k-1} C_i(x) \right] A(x), \quad N_A^0 C_i(x) = C_i(x) \quad (12)$$

$$d_i = \min \left\{ j : \left[\frac{\partial}{\partial x} N_A^{j-1} C_i(x) \right] B(x) \neq 0^T, j = 1, 2, \dots, n \right\}$$

számításokat és bevezetve a $C_{12} \cos(x_1 + x_2)$, $S_{12} = \sin(x_1 + x_2)$ jelölést kapjuk, hogy

$$N_A^0 C_1(x) = x_5, \quad N_A^1 C_1(x) = x_4 C_{12}, \quad N_A^2 C_1(x) = -\frac{1}{m_v} [T(S_{12}x_1 + C_{12}) + S_{12}S_h]$$

$$N_A^0 C_2(x) = x_6, \quad N_A^1 C_2(x) = x_4 S_{12}, \quad N_A^2 C_2(x) = \frac{1}{m_v} [T(C_{12}x_1 - S_{12}) + C_{12}S_h]$$

Az approximált nemlineáris rendszer differenciális rendje $d_1 = d_2 = 2$, továbbá teljesül

$$C^*(x) = \begin{bmatrix} N_A^{d_1} C_1(x) \\ N_A^{d_2} C_2(x) \end{bmatrix} = \begin{bmatrix} -\frac{1}{m_v} [T(S_{12}x_1 + C_{12}) + S_{12}S_h] \\ \frac{1}{m_v} [T(C_{12}x_1 - S_{12}) + C_{12}S_h] \end{bmatrix}_{2 \times 1}$$

$$S(x) = \begin{bmatrix} \frac{-S_{12}}{m_v} & \frac{S_{12}x_1 + C_{12}}{m_v} \\ \frac{C_{12}}{m_v} & \frac{-C_{12}x_1 + S_{12}}{m_v} \end{bmatrix}_{2 \times 2} \quad \text{és} \quad \det(S(x)) = -\frac{1}{m_v^2} \quad (13)$$

$$S^{-1}(x) = m_v \begin{bmatrix} C_{12}x_1 - S_{12} & S_{12}x_1 + C_{12} \\ C_{12} & S_{12} \end{bmatrix}_{2 \times 2}$$

$$M^*(x) = \begin{bmatrix} \alpha_{01} N_A^0 C_1(x) + \alpha_{11} N_A^1 C_1(x) \\ \alpha_{02} N_A^0 C_2(x) + \alpha_{12} N_A^1 C_2(x) \end{bmatrix} = \begin{bmatrix} \alpha_{01} x_5 + \alpha_{11} x_4 C_{12} \\ \alpha_{02} x_6 + \alpha_{12} x_4 S_{12} \end{bmatrix}_{2 \times 1}$$

Ezért a nemlineáris rendszerek Lie-algebrán alapuló eredményei szerint választható

$$u = S^{-1}(x) \left\{ -C^*(x) + \Lambda w - M^*(x) \right\} \quad (14)$$

ahol $\lambda = \text{diag}(\lambda_1, \lambda_2)$. Vezessük be a következő jelöléseket:

$$\begin{aligned} \bar{y}_1 &= \lambda_1 w_1 - \alpha_{01} x_5 - \alpha_{11} x_4 C_{12} \\ \bar{y}_2 &= \lambda_2 w_2 - \alpha_{02} x_6 - \alpha_{12} x_4 S_{12} \end{aligned} \quad (15)$$

akkor átalakítások után kapjuk, hogy a zárt rendszerre teljesül:

$$\begin{aligned} u_1 &= -S_h + [(C_{12} x_1 - S_{12}) \bar{y}_1 + (S_{12} x_1 + C_{12}) \bar{y}_2] m_v \\ u_2 &= T + [C_{12} \bar{y}_1 + S_{12} \bar{y}_2] m_v \\ \dot{y}_1 &= x_4 C_{12} \\ \dot{y}_2 &= x_4 S_{12} \\ \ddot{y}_1 &= \dot{x}_4 C_{12} - x_4 S_{12} (\dot{x}_1 + \dot{x}_2) \\ \dot{x}_1 + \dot{x}_2 &= (-S_{12} \bar{y}_1 + C_{12} \bar{y}_2) / x_4 \\ \dot{x}_4 &= C_{12} \bar{y}_1 + S_{12} \bar{y}_2 \\ \ddot{y}_1 &= \bar{y}_1 = \lambda_1 w_1 - \alpha_{01} x_5 - \alpha_{11} x_4 C_{12} = \lambda_1 w_1 - \alpha_{01} y_1 - \alpha_{11} \dot{y}_1 \\ \ddot{y}_2 &= \bar{y}_2 = \lambda_2 w_2 - \alpha_{02} x_6 - \alpha_{12} x_4 S_{12} = \lambda_2 w_2 - \alpha_{02} y_2 - \alpha_{12} \dot{y}_2 \end{aligned} \quad (16)$$

3.4 Nemlineáris kimeneti visszacsatoláson alapuló irányítás

Választható $\lambda_1 = \lambda_2 := \lambda$, $\alpha_{01} = \alpha_{02} := \lambda$ és $\alpha_{11} = \alpha_{12} = 2\sqrt{\lambda}$, amellyel két szétcsatolt aperiodikus határesetű másodrendű rendszerhez jutunk, amelynek karakterisztikus egyenlete $s^2 + 2\sqrt{\lambda}s + \lambda = 0$.

$$\ddot{y}_i + \alpha_{1i} \dot{y}_i + \alpha_{0i} y_i = \lambda_i w_i \quad (17)$$

Választható ezért $w_i := w_{ia} + \frac{1}{\lambda} (\alpha_{1i} \dot{w}_{ia} + \ddot{w}_{ia})$, amely után a két szétcsatolt rendszer

$$\begin{aligned} \ddot{y}_i + \alpha_{1i} \dot{y}_i + \lambda y_i &= \lambda \left[w_{ia} + \frac{1}{\lambda} (\alpha_{1i} \dot{w}_{ia} + \ddot{w}_{ia}) \right] \Rightarrow \\ (\ddot{w}_{ia} - \ddot{y}_i) + \alpha_{1i} (\dot{w}_{ia} - \dot{y}_i) + \lambda (w_{ia} - y_i) &= 0 \end{aligned} \quad (18)$$

alakú stabil rendszer lesz, ahol a referencia jelek szerepét betöltő $w_{1a} = X_a(t)$ és $w_{2a} = Y_a(t)$ a megtervezett ütközést elkerülő pályák.

A differenciálgeometriai elven alapuló irányításhoz szükség van a pálya első és második deriváltjára is, amelyet spline-technikával korábban már megterveztünk.

Vegyük észre, hogy az S_v irányítás és az állapotváltozók (vagy a becült állapotváltozók) ismeretében a valódi irányítás, vagyis a δ_w kormányzási szög meghatározható:

$$\delta_w = \frac{S_v}{c_F} + \beta + \frac{l_F \dot{\psi}}{v_G} \quad (19)$$

4. A nemlineáris prediktív irányítási algoritmus

A nemlineáris modellalapú prediktív irányítás egy lehetséges megvalósítása a mozgó horizontú irányítás (Receding Horizon Control, RHC), amely minden horizont kezdetén linearizálja a nemlineáris rendszert, és az így keletkező lineáris időinvariáns (LTI), vagy lineáris időben változó (LTV) rendszert optimalizálja. Az optimalizálási feladat egy költségfüggvény minimalizálása felnyitott körben a rendszer jövőbeli viselkedésének jóslására (predikciójára) alapozva [5]. Meghatározásra kerül az optimális beavatkozó jel (control) sorozat a horizonton belül, és kiadásra kerül a sorozat első eleme aktuális beavatkozó jelként zárt körben. Ez a lépés ciklikusan ismétlődik az új horizontra, amely az előzőnek T mintavételi idővel való eltolásával keletkezik. Nyitott kérdés a zárt nemlineáris rendszer stabilitása, amelyre jó hatással van a horizont N szélességének növelése (a horizont szélessége időben mérve NT , ahol T a szabályozás mintavételi ideje).

Ha a nemlineáris dinamikus modellt használnánk a predikcióra, akkor egy nemlineáris optimalizálási feladat keletkezne, amelynek valós idejű megoldása gyors rendszerek esetén idő-kritikus. Ezért a nemlineáris modell linearizálását választottuk az előírt nominális trajektória mentén minden horizont kezdetén, amelyet követ a perturbációs hatás optimalizálása a horizonton belül kvadrátikus kritérium szerint és analitikusan kezelhető végfeltétel mellett.

Komoly problémát jelent azonban, hogy a jármű alulaktuált, azaz nagyobb a szabadságfoka, mint a beavatkozó jelek száma, következésképp nem minden nominális pályához létezik azt pontosan megvalósító nominális irányítás. Ezért a megtervezett CAS akadályelkerülő pályához sem határozható meg egyszerű numerikus technikával a megfelelő approximációt biztosító nominális beavatkozó jel sorozat. Legfeljebb egy approximáló irányítást tételezhetünk fel, amelynek hatására a mozgás a nominális pálya közelében halad.

4.1 Választható alternatívák

Két lehetőség kínálkozik: i) LTI rendszer választása, amelyhez csak (x_0, u_0) szükséges, nem pedig a teljes $u(t)$ időfüggvény a horizonton belül. Ekkor u_0 választható az előző horizont optimális sorozata második elemének, ezért a fenti probléma nem jelentkezik, de a jóslás alapjául szolgáló LTI modell csak pontatlanabb jóslást tesz lehetővé. ii) LTV rendszer generálása a később ismertető algoritmus szerint. Ehhez abból indulhatunk ki, hogy ha a hiba a horizonton belül kicsi, akkor a horizonton belüli optimális beavatkozó jel sorozat a megtervezett nominális CAS pályához szükséges beavatkozó jel sorozat egy jó approximációjának tekinthető. Ezért linearizáljuk a rendszert az előző horizonton belül kapott optimális $x(t), u(t)$ mentén az új horizont kezdetén, és optimalizáljuk az ezek körüli perturbációt az új horizonton belül. Vegyük azonban észre, hogy az előző horizonton belüli optimális beavatkozó jel sorozat balra tolása miatt az új u_{N-1} a horizont végén hiányzik, ezért ennek számítását is ki kell dolgozni. A programban mindkét eset kiválasztható.

A továbbiakban az egyszerűség kedvéért *nominálisnak* nevezzük az előző horizontban keletkezett optimális, majd eltolt és kiegészített beavatkozó jel sorozatot, és ezzel szemben a *kívánt* (desired) beavatkozó jel sorozat az lesz, amely az LTV linearizált rendszer körüli perturbációk hatását minimalizálja. A módszer természetesen LTI rendszer esetére is alkalmazható, mivel az egyszerűbben képezhető LTI approximáció prediktív irányítási szempontból az LTV rendszer irányítása speciális esetének tekinthető (konstans rendszer a horizonton belül).

4.2 A megvalósított prediktív irányítás elméleti alapjai

Jelölje $\{x_0, x_1, \dots, x_N\}$ és $\{u_0, u_1, \dots, u_{N-1}\}$ a nominális állapot és bemenő (beavatkozó) jel sorozatot a horizonton belül, legyen továbbá \hat{x}_0 a becsült állapot a horizont kezdetén és $\{y_0 = Cx_0, y_1 = Cx_1, \dots, y_N = Cx_N\}$ az állapot sorozathoz tartozó kimenő jel sorozat. Legyen $\{y_{d0}, y_{d1}, \dots, y_{dN}\}$ a megkívánt (desired) kimenő jel sorozat (vagyis a megtervezett CAS referencia jel sorozat a horizonton belül) és jelölje az ehhez képesti hiba jel sorozatot $\{e_0 = y_{d0} - y_0, e_1 = y_{d1} - y_1, \dots, e_N = y_{dN} - y_N\}$. A nominális bemenő jel sorozat lehet a differenciálgeometriai elvű algoritmus (DGA) szerint számított irányítás a legelső horizont esetén, vagy a továbbiakban az előző optimális bemenő jel sorozat eggyel eltolva és egy új elemmel

kiegészítve, amely pl. egy végfeltételből számítható vagy az utolsó bemenő jel egyszerű megismétlése (lásd később az 1. lépést az algoritmusban).

A nemlineáris dinamikus modell linearizálható a nominális állapot és bemenő jel sorozatok mentén, és tekinthetők a keletkező lineáris időben változó (LTV) rendszer körüli $\delta x_0 = \hat{x}_0 - x_0$, $\delta x_1, \dots, \delta x_N$, $\delta u_0, \dots, \delta u_{N-1}$ perturbációk. A linearizálás történhet a pontos és az approximált (input affín) nemlineáris modell körül. A perturbációkat a $\delta x_{i+1} = A_i \delta x_i + B_i \delta u_i$ LTV rendszer írja le. A kimeneti hibasorozat $y_{di} - C(x_i + \delta x_i) = e_i - \delta y_i$, a J költségfüggvény pedig választható egy kvadratikus függvénynek, amely bünteti a kimeneti hibákat és a névleges irányítástól való nagy eltéréseket:

$$J = \frac{1}{2} \sum_{i=1}^{N-1} \|e_i - \delta y_i\|^2 + \frac{1}{2} \lambda \sum_{i=0}^{N-1} \|\delta u_i\|^2 \quad (20)$$

Az állapot és kimenő jel perturbációk a következőképp számíthatók:

$$\begin{pmatrix} \delta x_1 \\ \delta x_2 \\ \vdots \\ \delta x_{N-1} \\ \delta x_N \end{pmatrix} = \begin{bmatrix} A_0 \\ A_1 A_0 \\ \vdots \\ A_{N-2} \cdots A_1 A_0 \\ A_{N-1} \cdots A_1 A_0 \end{bmatrix} \delta x_0 + \begin{bmatrix} B_0 & 0 & \cdots & 0 & 0 \\ A_1 B_0 & B_1 & \cdots & 0 & 0 \\ \vdots & \cdots & \ddots & 0 & 0 \\ A_{N-2} \cdots A_1 B_0 & A_{N-2} \cdots A_2 B_1 & \cdots & B_{N-2} & 0 \\ A_{N-1} \cdots A_1 B_0 & A_{N-1} \cdots A_2 B_1 & \cdots & A_{N-1} B_{N-2} & B_{N-1} \end{bmatrix} \begin{pmatrix} \delta u_0 \\ \delta u_1 \\ \vdots \\ \delta u_{N-2} \\ \delta u_{N-1} \end{pmatrix} \quad (21)$$

$$\begin{pmatrix} \delta y_1 \\ \delta y_2 \\ \vdots \\ \delta y_{N-1} \\ \delta y_N \end{pmatrix} = \begin{bmatrix} CA_0 \\ CA_1 A_0 \\ \vdots \\ CA_{N-2} \cdots A_1 A_0 \\ CA_{N-1} \cdots A_1 A_0 \end{bmatrix} \delta x_0 + \begin{bmatrix} CB_0 & 0 & \cdots & 0 & 0 \\ CA_1 B_0 & CB_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ CA_{N-2} \cdots A_1 B_0 & \cdots & \cdots & CB_{N-2} & 0 \\ CA_{N-1} \cdots A_1 B_0 & \cdots & \cdots & CA_{N-1} B_{N-2} & CB_{N-1} \end{bmatrix} \begin{pmatrix} \delta u_0 \\ \delta u_1 \\ \vdots \\ \delta u_{N-2} \\ \delta u_{N-1} \end{pmatrix} \quad (22)$$

vagy tömör alakban

$$\begin{pmatrix} \delta y_1 \\ \delta y_2 \\ \vdots \\ \delta y_{N-1} \end{pmatrix} = P_1 \delta x_0 + H_1 \delta U; \quad \delta y_N = P_2 \delta x_0 + H_2 \delta U \quad (23)$$

$$P_1 = \begin{bmatrix} p_1^T \\ \vdots \\ p_{N-1}^T \end{bmatrix}_{m(N-1) \times n}, \quad H_1 = \begin{bmatrix} h_1^T \\ \vdots \\ h_{N-1}^T \end{bmatrix}_{m(N-1) \times Nr}, \quad P_2 = [p_N^T]_{m \times n}, \quad H_2 = [h_N^T]_{m \times Nr}$$

ahol $n = \dim x$, $r = \dim u$, $m = \dim y$.

Az optimalizálási probléma végfeltétellel a következő alakú:

$$J = \frac{1}{2} \sum_{i=1}^{N-1} \|e_i - \delta y_i\|^2 + \frac{1}{2} \lambda \sum_{i=0}^{N-1} \|\delta u_i\|^2 \rightarrow \min \quad (24)$$

$$e_N - \delta y_N = e_N - (P_2 \delta x_0 + H_2 \delta U) = 0$$

A (23) jelölésekkel a költségfüggvény részletes alakja:

$$J = \frac{1}{2} \sum_{i=1}^{N-1} \langle e_i, e_i \rangle - \langle (\sum_{i=1}^{N-1} p_i e_i), \delta x_0 \rangle - \langle (\sum_{i=1}^{N-1} h_i e_i), \delta U \rangle + \frac{1}{2} \langle (\sum_{i=1}^{N-1} p_i p_i^T) \delta x_0, \delta x_0 \rangle \quad (25)$$

$$+ \langle (\sum_{i=1}^{N-1} h_i p_i^T) \delta x_0, \delta U \rangle + \frac{1}{2} \langle (\sum_{i=1}^{N-1} h_i h_i^T) \delta U, \delta U \rangle + \frac{1}{2} \lambda \langle \delta U, \delta U \rangle.$$

Mivel a költségfüggvény konvex és a korlátozás lineáris, ezért az optimum szükséges és elégséges feltétele a Lagrange multiplikátor szabály. Jelölje a μ vektor a Lagrange-multiplikátorokat, akkor

$$\begin{aligned} L &= J + \langle \mu, e_N \rangle - \langle P_2^T \mu, \delta x_0 \rangle - \langle H_2^T \mu, \delta U \rangle \\ 0 &= \frac{dL}{d\delta U} = -\sum_{i=1}^{N-1} h_i e_i + \left(\sum_{i=1}^{N-1} h_i p_i^T \right) \delta x_0 + \left(\sum_{i=1}^{N-1} h_i h_i^T \right) \delta U + \lambda \delta U - H_2^T \mu \\ &= -\bar{m} + H_1^T P_1 \delta x_0 + (H_1^T H_1 + \lambda I) \delta U - H_2^T \mu \end{aligned} \quad (26)$$

ahol $\bar{m} = \sum_{i=1}^{N-1} h_i e_i$. A következő jelölésekkel az eredmények egyszerűbb alakban írhatók fel:

$$L_1 := H_1^T H_1 + \lambda I, \quad L_\mu := H_2 L_1^{-1} H_2^T \quad (27)$$

Ekkor (26) és a végfeltétel korlátozás figyelembevételével:

$$\begin{aligned} \delta U &= L_1^{-1} (\bar{m} + H_2^T \mu - H_1^T P_1 \delta x_0) \\ e_N - P_2 \delta x_0 - H_2 [L_1^{-1} (\bar{m} + H_2^T \mu - H_1^T P_1 \delta x_0)] &= 0 \\ \mu &= L_\mu^{-1} [e_N - H_2 L_1^{-1} \bar{m} - (P_2 - H_2 L_1^{-1} H_1^T P_1) \delta x_0] \end{aligned}$$

és ezért

$$\begin{aligned} \delta U &= L_1^{-1} \{ H_2^T L_\mu^{-1} e_N + (I - H_2^T L_\mu^{-1} H_2 L_1^{-1}) \bar{m} \\ &\quad - [H_1^T P_1 + H_2^T L_\mu^{-1} (P_2 - H_2 L_1^{-1} H_1^T P_1)] \delta x_0 \} \end{aligned} \quad (28)$$

A beavatkozó jel zárt körben $u_0 + \delta u_0$, ahol u_0 a nominális irányítás a horizont kezdetén és $\delta u_0 \in R^r$ a felnyitott körben optimális δU sorozat első eleme.

4.3 A megvalósított prediktív irányítási algoritmus:

Minden horizontban a következő lépések ismétlődnek:

1. lépés. Az x_0 kezdeti állapotból és az $\{u_0, u_1, \dots, u_{N-1}\}$ nominális bemenő jel sorozatból meghatározásra kerül az $\{x_0, x_1, \dots, x_N\}$ nominális állapot sorozat a jármű approximált nemlineáris dinamikus modellje alapján. Itt x_0 az eltolt előző horizontból jön, és eltérhet a becsült \hat{x}_0 kezdeti állapottól.

Az előírt (desired) állapot sorozat a megtervezett CAS akadályelkerülő pályából számítható nulla β oldalirányú elcsúszási szög (slide slip angle) esetén. A kimenő jel $y = (X, Y)^T$, ezért a kívánt (desired) kimenő jel sorozat kiszámítható a kívánt (desired) állapot sorozatból a horizontban a $C = [e_5 \ e_6]^T$ mátrix segítségével, ahol kivételesen e_i az i -edik standard egységvektort jelöli. A hibajel sorozat ezek különbsége. A legelső horizont esetén a nominális bemenő jel sorozat a DGA módszerrel kerül meghatározásra és x_0 inicializált értéke a megtervezett CAS akadályelkerülő pálya állapotvektora nulla oldalcsúszási szög (slide slip angle) esetén.

2. lépés. A diszkrétidejű $\delta x_{i+1} = A_i \delta x_i + B_i \delta u_i$ LTV rendszer meghatározása a $\dot{x} = f_c(x, u)$ folytonosidejű approximált nemlineáris dinamikus modelltől Euler-formulával:

$$A_i := I + T df_c / dx|_{(x_i, u_i)}, \quad B_i := T df_c / du|_{(x_i, u_i)}$$

3. lépés. Az optimális δU bemenő jel változás sorozat meghatározása (28) alapján és $\delta x_0 = \hat{x}_0 - x_0$ felhasználásával, ahol \hat{x}_0 a becsült állapot. Az optimális bemenő jel sorozat $U := U + \delta U$. A sorozat első u_0 eleme kerül kiadásra beavatkozó jelként zárt szabályozási körben.

4. lépés. Azért, hogy inicializálható legyen a bemenő jel sorozat a következő horizont számára, az x_0 kezdeti állapothoz és az új $\{u_0, u_1, \dots, u_{N-1}\}$ optimális bemenő jel sorozathoz az $\dot{x} = f_c(x, u)$ folytonosidejű approximált nemlineáris dinamikus modell felhasználásával meghatározásra kerül a nemlineáris rendszer állapot sorozata, az eredményt a tranziens végén x_N jelöli.

Az ismeretlen u_N háromféleképpen határozható meg: i) u_N a DGA módszerrel kerül meghatározásra x_N felhasználásával. ii) u_N úgy lesz megválasztva, hogy az Euler-formulával képzett diszkrétidejű nemlineáris rendszer x_{N+1} válasza és a megtervezett CAS akadályelkerülő pálya $x_{d,N+1}$ állapota közötti differencia, amely $f(x_N) + G(x_N)u_N - x_{d,N+1}$, legyen minimális LS (least squares) értelemben. iii) Az utolsó bemenő jel az optimális bemenő jel sorozatban egyszerűen ismétlésre kerül: $u_N := u_{N-1}$.

5. lépés. A nominális bemenő jel sorozat a következő horizont számára $\{u_1, u_2, \dots, u_N\}$, amely a kiegészített optimális bemenő jel sorozat $\{u_0, u_1, \dots, u_N\}$ eggyel balra eltolva.

4.4 Integrátor beillesztése a szabályozóba

A követési tulajdonságok további javítása érdekében lehetséges integrátor beillesztése a prediktív szabályozóba a $\delta x_i := (\delta x_i^T, \delta u_{i-1}^T)^T$ bővített (augmented) állapot bevezetésével, ahol $\delta u_i = \delta u_{i-1} + \delta r_i$, ekkor azonban a bemenő jel δr_i változását kell optimalizálni. Bevezetve a

$$A_i := \begin{bmatrix} A_i & B_i \\ 0 & I \end{bmatrix} \text{ és } B_i := \begin{bmatrix} B_i \\ I \end{bmatrix} \quad (29)$$

helyettesítéseket, a korábbi eredmények az új változókkal érvényben maradnak. Mindazonáltal ebben az esetben δR lesz az optimális bemenő jel változás sorozat, amely meghatározásra kerül, és az optimális δU bemenő jel sorozat a kumulatív összege δR -nek.

5. GPS/INS jeleken és kétszintű Kalman-szűrőn alapuló állapotbecslés

A reális rendszerekben, így járművekben is, az állapotváltozók többsége nem mérhető, ezért becslésüket az érzékelők mérhető jeleire kell alapozni. Feltételeztük, hogy az érzékelők jeleit 2-antennás differenciális GPS rendszer jelei és egy INS rendszer (Inertial Navigation System) gyorsulási érzékelő/gyroszkóp jelei szolgáltatják.

A GPS/INS jelek kiértékelését [3] eredményeire alapoztuk. A berendezések a drivereik segítségével magas szintű, állapotbecslésre alkalmas információt szolgáltatnak, amelyek a következők:

V_m^{GPS} : a jármű (mért, erre utal az m-index) sebességvektora a GPS földi inerciarendszerben

ψ_m^{GPS} : a jármű orientációja a GPS földi inerciarendszerben

$a_{x,m}$: a jármű x -irányú (longitudinális) gyorsulása a járműhöz rögzített koordináta-rendszerben

$a_{y,m}$: a jármű y -irányú (transzverzális) gyorsulása a járműhöz rögzített koordináta-rendszerben

r_m : a jármű z -irányú (yaw-irányú) szögsebessége a járműhöz rögzített koordináta-rendszerben

Feltesszük, hogy az első antenna közvetlenül az INS rendszer felett helyezkedik el, és az INS rendszer a tömegközéppontban (COG) van. (Az általános esetben változtatások szükségesek.)

Az érzékelők drift-tel (bias) is rendelkeznek, amelyeket a Kalman-szűrőknek szintén becsülni kell. Az érzékelők pontosságát a σ_1 érték és a bias jellemzi. Az additív Gauss-zaj (noise) lényegében a mért érték körüli $[-3\sigma, 3\sigma]$ sávba esik statisztikailag, amelyhez még hozzáadódik a bias értéke.

Vízszintes síkban haladó jármű esetén $V_m^{GPS} = (V_1^{GPS}, V_2^{GPS}, 0)^T$ alakú, ahonnan meghatározható az oldalcsúszási szög mért értéke, abból pedig a járműhöz rögzített koordináta-rendszerben az u sebességvektor komponensei:

$$\begin{aligned}
\gamma &= \text{atan2}(V_2^{GPS}, V_1^{GPS}) \Rightarrow \beta^{GPS} = \gamma - \psi_m^{GPS} \\
u_{x,m}^{GPS} &= \|V^{GPS}\| \cos(\beta^{GPS}) + \text{noise} \\
u_{y,m}^{GPS} &= \|V^{GPS}\| \sin(\beta^{GPS}) + \text{noise}
\end{aligned} \tag{30}$$

Az állapotbecslést 2 fokozatból álló Kalman-szűrőre alapozzuk. Az első fokozat becsli a $\dot{\psi} = r$ szögsebességet. Erre a következő két módszer valamelyike javasolható:

$$\begin{aligned}
\begin{pmatrix} \dot{\psi} \\ \dot{r}_{bias} \end{pmatrix} &= \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} \psi \\ r_{bias} \end{pmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} r_m + \text{noise} \\
\psi_m^{GPS} &= [1 \quad 0] \begin{pmatrix} \psi \\ r_{bias} \end{pmatrix} + \text{noise}
\end{aligned} \tag{31}$$

$$\begin{aligned}
\frac{d}{dt} \begin{pmatrix} \psi \\ 1/s_r \\ r_{bias}/s_r \end{pmatrix} &= \begin{bmatrix} 0 & r_m & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} \psi \\ 1/s_r \\ r_{bias}/s_r \end{pmatrix} + \text{noise} \\
\psi_m^{GPS} &= [1 \quad 0 \quad 0] (\psi \quad 1/s_r \quad r_{bias}/s_r)^T + \text{noise}
\end{aligned} \tag{32}$$

ahol s_r a giroszkóp érzékenysége. Az állapotbecslést diszkrétidejű Kalman-szűrő végzi, ehhez a választott folytonosidejű modellről diszkrétidejűre kell áttérni. Jelölje T a mintavételi időt, akkor a diszkrétidejű modellek mátrixai a következők:

$$A_{d1} = \begin{bmatrix} 0 & -T \\ 0 & 1 \end{bmatrix}, \quad B_{d1} = \begin{bmatrix} T \\ 0 \end{bmatrix}, \quad C_{d1} = [1 \quad 0] \tag{33a}$$

$$A_{d1} = I_3 + A_{c1}T, \quad B_{d1} = \mathbf{0}_{3 \times 1}, \quad C_{d1} = [1 \quad 0 \quad 0] \tag{33b}$$

Az állapotbecslés eredményéből a szögsebesség becsült értéke kifejezhető a választott modell alapján:

$$r = \hat{\dot{\psi}} := -\hat{r}_{bias} + r_m \tag{34a}$$

$$r = \hat{\dot{\psi}} := r_m (1/\hat{s}_r) - (\hat{r}_{bias}/\hat{s}_r) \tag{34b}$$

Az így keletkező érték felhasználásra kerül a második Kalman-szűrőben a jármű u_x, u_y sebességének becslésekor. A becslés alapja az $a = \dot{u} + \omega \times u$ összefüggés, amely a következő folytonosidejű modellt eredményezi:

$$\begin{aligned}
\frac{d}{dt} \begin{pmatrix} u_x \\ a_{x,bias} \\ u_y \\ a_{y,bias} \end{pmatrix} &= \begin{bmatrix} 0 & -1 & r & 0 \\ 0 & 0 & 0 & 0 \\ -r & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} u_x \\ a_{x,bias} \\ u_y \\ a_{y,bias} \end{pmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} a_{x,m} \\ a_{y,m} \end{pmatrix} + \text{noise} \\
\begin{pmatrix} u_{x,m}^{GPS} \\ u_{y,m}^{GPS} \end{pmatrix} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} u_x \\ a_{x,bias} \\ u_y \\ a_{y,bias} \end{pmatrix} + \text{noise}
\end{aligned} \tag{35}$$

A modell a valós időben változó r értékét tartalmazza, ezért előnyös, hogy a folytonos időről diszkrét időre áttérés eredménye analitikusan megadható:

$$\begin{aligned}
A_{d2} &= \begin{bmatrix} \cos(rT) & -\sin(rT)/r & \sin(rT) & -[1-\cos(rT)]/r \\ 0 & 1 & 0 & 0 \\ -\sin(rT) & [1-\cos(rT)]/r & \cos(rT) & \sin(rT)/r \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
B_{d2} &= \begin{bmatrix} \sin(rT)/r & [1-\cos(rT)]/r \\ 0 & 0 \\ -[1-\cos(rT)]/r & \sin(rT)/r \\ 0 & 0 \end{bmatrix} \\
C_{d2} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}
\end{aligned} \tag{36}$$

Mivel a gyakorlati esetben $r \approx 0$ nem kizárható, ezért a kritikus határértékeket a L'Hospital-szabállyal analitikusan határoztuk meg a szoftver implementáció számára. Az állapotbecslést a második Kalman-szűrő végzi. A becslésből kapott \hat{u}_x, \hat{u}_y és az első Kalman-szűrő eredményeként kapott $\hat{\psi}$ értékéből további állapotváltozók becslése határozható meg:

$$\hat{v}_G = \sqrt{\hat{u}_x^2 + \hat{u}_y^2}; \quad \hat{\beta} = \text{atan2}(\hat{u}_y, \hat{u}_x) \tag{37}$$

A hiányzó X, Y pozíció jellegű állapotváltozókat a Kalman-szűrőkkel kapott becslt értékekből téglalap szabályon alapuló numerikus integrálással határozzuk meg:

$$\hat{X} := \hat{X} + T\hat{v}_G \cos(\hat{\psi} + \hat{\beta}); \quad \hat{Y} := \hat{Y} + T\hat{v}_G \sin(\hat{\psi} + \hat{\beta}) \tag{38}$$

A Kalman-szűrőket az alábbi alakban implementáltuk:

$$\begin{aligned}
x_-(t+1) &= A_d x_-(t) + B_d u(t) \\
P_-(t+1) &= A_d P_+(t) A_d^T + Q
\end{aligned} \tag{39a}$$

(time update)

$$\begin{aligned}
x_+(t) &= x_-(t) + K[y(t) - Cx_-(t)] \\
K &= P_-(t)C^T [CP_-(t)C^T + R]^{-1} \tag{39b} \\
P_+(t) &= [I - KC]P_-(t)
\end{aligned}$$

(measurement update)

A szenzormérések tipikus mintavételi ideje $T_{INS} = T = 0.01s$ (100Hz), $T_{GPS,vel} = 0.1s$ (10Hz), $T_{GPS,aut} = 0.2s$ (5Hz), ahol T a szabályozás mintavételi ideje. Az eltérő mintavételi időket az állapotbecsléskor figyelembe kell venni.

6. A szabályozásokat megvalósító stand-alone program

A szabályozásokat megvalósító program különböző szabályozások vizsgálatára alkalmas, amelyek közül a differenciálgeometriai elvű irányítás és a nemlineáris prediktív irányítás került kifejlesztésre. A program azonban lehetővé teszi más irányítások későbbi beillesztését is. A gyakorlatban elterjedt Matlab Simulink→Real Time Workshop→Target Compiler irányzat elsősorban egyszerű, végtelen ciklusban futó szabályozások leképezésére alkalmas, a korszerű irányítások ezzel szemben különféle programállapotokat különböztetnek meg, melyek csak pótlólagos eszközök (State Flow stb.) bevonásával lennének megvalósíthatók. Ezért a szoftver megvalósításakor kerültük a Simulink és a MATLAB toolboxok használatát annak érdekében, hogy a későbbi C-nyelvű megvalósításokor alkalmazhatók legyenek a MATLAB Compiler szolgáltatásai, kibővítve speciális C-nyelvű

programokkal (periféria driver, feladat orientált könyvtárak, GUI). Ehhez tudni kell, hogy a MATLAB egy interpretált nyelv, amelynek használatakor a MATLAB, Simulink, Java interpreterek futnak, és ezek más futtató környezetben már nem állnak rendelkezésre, ennek következtében a MATLAB Compiler szolgáltatásai is erősen korlátozottak.

6.1 Az automatikus akadályelkerülést megvalósító frameCAS stand-alone program

A kutatás során először meghatároztuk a rendelkezésre álló két legfejlettebb termék, a Matlab 6.5 (R13) Compiler 3 és a Matlab R2006a (R14) Compiler 4 korlátait stand-alone programok létrehozása során.

Megállapítottuk, hogy a Matlab 6.5 Compiler Version 3 C/C++ nyelvre tudja a Matlab függvényeket lefordítani, amely azután stand-alone programmá összeszerkeszthető, de a Matlab toolboxok nem, vagy csak lényeges megszorításokkal használhatók. Mivel a perspektivikus célok előnyben részesítik a C/C++ nyelvet a továbbfejlesztések során, ezért részletesen analizáltuk a fordított a korlátok tekintetében. A vizsgálatok eredményeképpen szükség volt a feladat szempontjából kulcsfontosságú Optimization Toolbox lecserélésére egy saját fejlesztésű változatra a korlátok felszámolása érdekében.

A szoftver technológiailag fejlettebb Matlab R2006a Compiler 4 nem C/C++ nyelvre, hanem a felhasználó számára nem definiált és üzleti okokból kriptografikus kulcsokkal védett közbenső CTF formátumra fordít, amely függ a futtató rendszertől, és amelyet azután a stand-alone program az indításkor automatikusan értelmez és végrehajt. A Matlab Optimization Toolbox szolgáltatásai lefordíthatók, a korlátok az alkalmazás szempontjából nem kritikusak, a probléma a C/C++ kimenet hiánya. A Compiler 4 biztosítja osztott könyvtárak (shared library target) szolgáltatásainak beépítését a stand alone programba.

Az így szerzett tapasztalatokra alapozva lehetőség nyílt stand-alone programok kifejlesztésére az automatikus akadályelkerülő pálya meghatározására és irányítással történő megvalósítására [6]. Ennek során figyelembe vettük a Matlab Compiler megismert korlátait, és a szükséges mértékben átdolgozzuk a korábbi programverziót a stand-alone korlátok figyelembevételével. A program elvárt bemeneti adatait a statikus akadály (pl. elől haladó járműről leesett teher) és a mozgó akadály (szembejövő jármű) geometriai paraméterei (befoglaló kör középpontja és sugara) alkotják, valamint a mozgó akadály és a saját jármű sebessége. További bemeneti adatok azonosítják az útszakaszt (bal oldali és jobb oldali sáv szélesség). Az akadályelkerülő pályát az elasztikus szalag elvére épülve határozzuk meg, amely nagyméretű nemlineáris egyenletrendszerre vezet, amelyet az Optimization Toolbox *fsolve* függvényével oldunk meg.

Az akadályelkerülő pályát differenciálgeometriai elvű és prediktív irányítási (mozgó horizontú, RHC) módszerekkel valósítjuk meg. Minden horizont kezdetén a rendszer meghatározza a mozgó jármű (időinvariáns vagy időben változó) linearizált modelljét az aktuális állapot vagy a teljes állapot-trajektória körül, és a prediktív irányítást a mozgó horizonton belül a keletkező LTI vagy LTV rendszerre alapozza. Az állapotbecslés GPS/INS mérésekre alapozott és kétszintű Kalman-szűrőt használ.

A szabályozásokat megvalósító saját fejlesztésű *frameCAS* keretprogram a bemeneti adatokat egy kötött struktúrájú, könnyen értelmezhető fájlban várja, köztük az akadályok geometriai és sebesség paramétereit, a saját jármű átlagsebességét és az elvégzendő funkciókat. A program kimeneti adatait a választott szabályozás melletti tranziensek alkotják, amelyek felrajzolásra kerülnek és fájlba kimenthetők. A program működése a következő lépésekből áll:

1. `initband()`, amely automatikusan megtervezi az akadályokat elkerülő CAS pályát
2. `kinematics()`, amely előállítja a szabályozások referencia jeleit
3. `ddc_euler()`, amely meghatározza a szabályozási rendszer tranzienseit RHC/DGA módszerrel
4. A szabályozási tranziensek grafikus megjelenítése

A keretprogramot és a meghívott függvényeket globális program switch-ek vezérlik, a programkapcsolók különféle kombinációkra adnak lehetőséget a program futásakor.

- a) `sys_appr`: arról dönt, hogy a tranziensek számítása az approximált nemlineáris vagy a pontos nemlineáris modell alapján történik-e (a szabályozó kimenetének számításakor mindig az approximált modellt használjuk)
- b) `sys_estim`: arról dönt, hogy a szabályozó kimenetének számításakor az állapotváltozókat, vagy azok állapotbecsléssel számított értékét használjuk-e

- c) `sys_contr`: arról dönt, mi az érvényes szabályozó, például a differenciálgeometriai (Lie-algebrai) elvekre épülő szabályozó esetén értéke a `'diffgeom'` string, nemlineáris prediktív irányítás esetén `'nonlinpred'` string.
- d) `int_horizon`: arról dönt, van-e integrátor a prediktív irányítás szerinti szabályozóban.

6.2 Szimulációs eredmények RHC irányítás esetén a stand-alone programmal

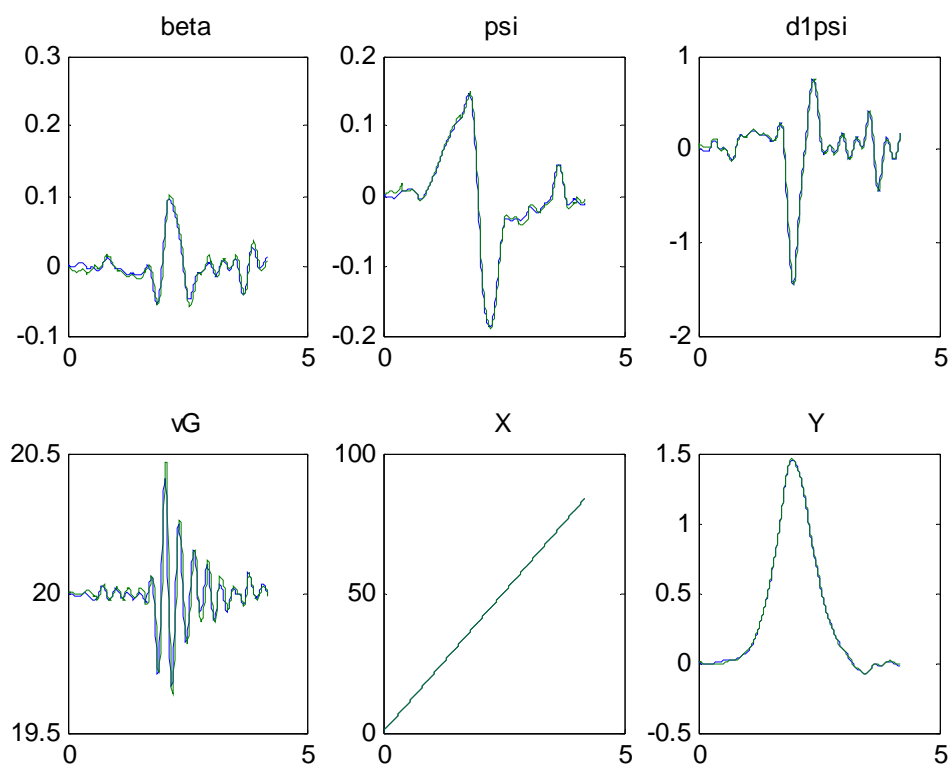
A bemeneti fájlban (`Syspar_File.txt`) az akadályok és a programkapcsolók értékeit a következőkre választottuk:

```

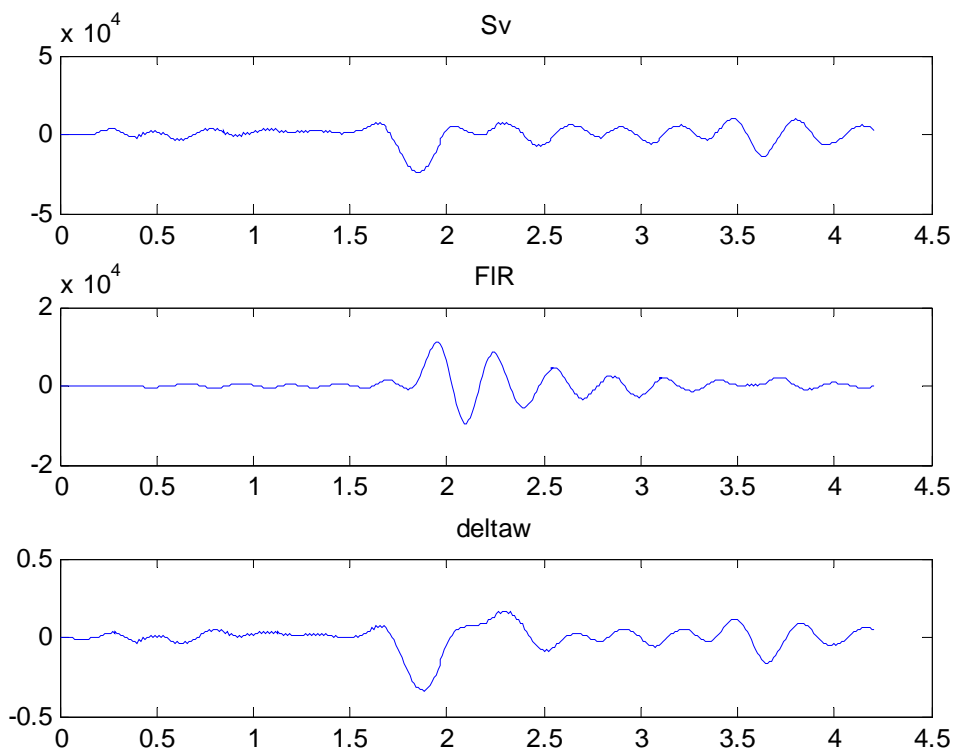
%*****
%Input file for CAS system parameters
%*****
%initband parameters
fv_own=20; %Own car average velocity
fstat_obs1=[40 0 2.5]; %[rx ry d] static_obstacle_1
fmov_obs=[120 3.5 4 15]; %[rx ry d v] moving_obstacle
froad_wide=[7 0.75 0.25]; %[total_wide left_portion right_portion]
%funframeCAS parameters
fsys_appr=0; %1->approximated_model_in_use
fsys_estim=1; %1->state_estimator_is_running
fsys_contr='nonlinpred'; %1->predictive_control_in_use
fdeltaw_horizon=0; %1->u(1)_is_deltaw, 0->u(1)_is_Sv_transversal
fdgfresh_horizon=1; %1->uN_by_diffgeom, 0->xNp1_to_0, 2->uN_to_uNm1
flambda_horizon=10; %lambda_weights_u_or_deltau_in_cost_function
fint_horizon=1; %1->integrator_in_RHC_controller
fLTV_horizon=1; %1->LTV_linearization_in_the_horizons

```

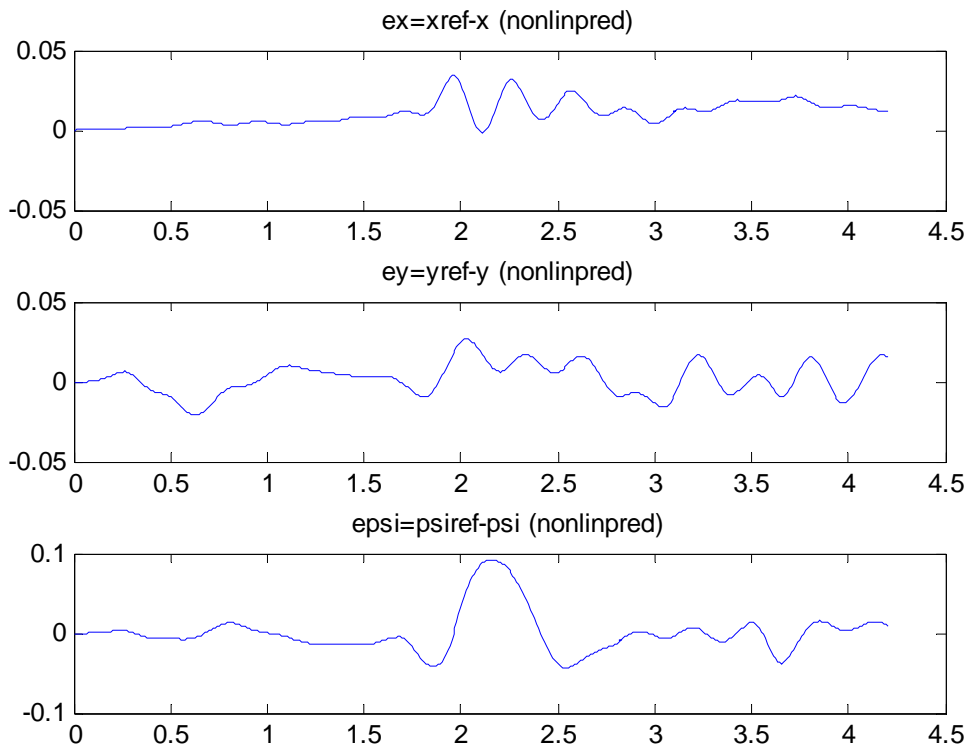
A stand-alone programmal (mindhárom verzióval) kapott szabályozási tranziensek eredményeit RHC irányítás és GPS/INS szenzorok esetén az 1-3. ábrák mutatják. Az ábrákon a tranziensek SI mértékegységekben értendők (m, rad, s, m/s, N stb.). A horizont hossza és a mintavételi idő rendre $N=10$ és $T=0.01s$.



1. ábra. Állapotváltozók és becslésük `sys_appr=0`, `sys_estim=1`, `sys_contr='nonlinpred'` esetén



2. ábra. Beavatkozó jelek $\text{sys_appr}=0$, $\text{sys_estim}=1$, $\text{sys_contr}='nonlinpred'$ esetén



3. ábra. Hibajelek $\text{sys_appr}=0$, $\text{sys_estim}=1$, $\text{sys_contr}='nonlinpred'$ esetén

7. Következtetések

A cikk egy javított módszert mutat be az automatikus akadályelkerülő pálya tervezésére ütközésselkerülő (CAS) rendszerben, amely feloldja az elasztikus szalagon alapuló pályatervezés hiányosságait. Az így kapott pálya simítására és a szabályozás referencia jelének meghatározására spline technikán alapuló módszert mutat be. Algoritmust ad az alulaktuált jármű modell alapú mozgó horizontú (RHC) nemlineáris prediktív irányítására, amely minden horizont kezdetén alkalmasan választott nominális beavatkozó jel és állapot sorozat mentén linearizálja a rendszert, meghatározza az így kapott időinvariáns (LTV) rendszer optimális beavatkozó jel sorozatát a referencia jelhez képesti hiba szerinti kvadratikus kritérium analitikus optimalizálásával felnyitott körben előírt végfeltétel, mint korlátozás mellett, és a sorozat első elemét ráadja a rendszer bemenetére zárt körben. A legelső horizontban a nominális beavatkozó jel sorozatot differenciálgeometriai módszeren (DGA) alapuló irányítással határozza meg a közelítő bemenet affín modell nemlineáris szétcsatlósásával és kompenzálásával. Az algoritmus három módszer alapján tudja frissíteni a nominális beavatkozó jel sorozatot a következő horizont számára. A nem mérhető állapotokat GPS/INS szenzor rendszer jeleinek bevonásával kinematikai modellen alapuló kétszintű Kalman-szűrővel határozza meg az irányítás számára, amely figyelembe veszi a GPS, gyorsulásérzékelő és giroszkóp eltérő mintavételi frekvenciáit. Az irányító rendszer szoftvere Matlab licenzzel nem igénylő stand-alone program, helyes működését a transziensek rendszerteknikai tulajdonságai (kis állapotbecslési hibák, technikailag megvalósítható nagyságrendű beavatkozó jelek és kis szabályozási hibák) demonstrálják.

Köszönetnyilvánítás

A kutatási eredmények az NKTH RET 04/2004 kutatási program RET 1.1 Járműforgalmi rendszerek modellezése és irányítása projekt támogatásával keletkeztek.

Irodalomjegyzék

1. Brandt, T., & Sattel, T. (2005). Path planning for automotive collision avoidance based on elastic bands. *Proceedings of the IFAC World Congress, Prague*, Paper TU-M16-TO/4 02746.pdf.
2. Börner, M., Andreani, L., Albertos, P., & Isermann, R. (2002). Detection of lateral vehicle driving conditions based on the characteristic velocity. *Proceedings of the IFAC World Congress, Barcelona, Conference CD*.
3. Ryu, J., & Gerdes, J.G. (2004) *Integrating Inertial sensors with Global Positioning System (GPS) for Vehicle Dynamic Control*. Journal of Dynamic Systems, Measurement, and Control. Vol. 126, pp. 242-254.
4. Freund, E., & Mayr, R. (1997) *Nonlinear Path Control in Automated Vehicle Guidance*. IEEE Transactions on Robotics and Automation, Vol. 13, pp. 49-60.
5. Lantos, B. (2006). *Algoritmusok gépjármű ütközésmentes pályatervezésére és pályakövetésére. Tanulmány*. BME Irányítástechnika és Informatika Tanszék, p.36. Készült a RET 1.1. Járműforgalmi rendszerek modellezése és irányítása projekt keretében.
6. Lantos, B. (2006). *Path Design and Receding Horizon Control for Collision Avoidance System of Cars*. WSEAS Transaction on Systems and Control, Issue 2, Vol. 1, pp. 105-112.