Modell Prediktív Kontroll Illesztése Járványügyi Modellhez

Epres Tamás

2024. július

1 Bevezetés

Az elmúlt néhány év rávilágított arra a tényre, hogy egy világjárvány irányítása és kézben tartása nem triviális feladat. Egy járvány lefolyásának irányítása összetett probléma, amelyhez több különböző tudományág területének ismerete nélkülözhetetlen; ebben az eseteben is olyan problémákba ütközhetünk, amelyhez szükséges a mérnöki problémamegoldás. Munkámban főleg szabályozástechnika szemszögéből fogom megvilágítani azokat a mérnöki megoldásokat és feladatokat, amelyek egy járvány irányításához szükségesek; ilyenek például a beavatkozó jel kvantáltsága és a beavatkozások hosszainak korlátozása. A feladat megoldása során modell prediktív kontrollt fogok alkalmazni különböző járványügyi modellekhez. Kezdetben bemutatom az alkalmazott stratégiát, a modell prediktív kontroll alapjait egy egyszerű lineáris modellen, majd a stratégiát alkalmazom egy ország járványügyi dinamikáját leíró folyamatmodelljén. Végül, de nem utolsó sorban, egy neurális hálóval mimikált multiágens modellhez szeretnék illesztetni modell prediktív kontrollt.

2 A Modell Prediktív Kontroll Bevezetése

2.1 A Modell Prediktív Kontroll Definíciója

Ebben a szekcióban főleg az irányítás elméleti megfontolásokkal, felmerülő problémákkal foglalkozok és nem a járványmodellel.A modell prediktív kontroll (későbbiekben az angol elnevezéssel utalva csak MPC) egy irányítástechnikai stratégia, módszer, amelynek lényege, hogy a szakaszt leíró dinamikát ismerve olyan beavatkozó szekvenciákat adjunk meg, amelyek egy bizonyos hibát minimalizálnak. Fogalmazhatunk úgy is, hogy az MPC egy adott időablakban egy hibát optimalizál, és az alapján ad meg egy beavatkozó jelet. Induljunk ki egy egyszerű lineáris diszkrét idõjú szakaszból:

$$\mathbf{x_{k+1}} = \mathbf{A}\mathbf{x_k} + \mathbf{B}u_k \tag{1}$$

$$\mathbf{y_k} = \mathbf{C}\mathbf{x_k} \tag{2}$$

Ahol:

- $\mathbf{y}_k$ a k-edik időpontban a szakasz válaszvektora.
- $\mathbf{x}_k$ a k-edik időpontban a szakasz állapotvektora.
- u_k a k-edik időpontban a beavatkozás.
- $\mathbf{A}$ a szakasz dinamikáját leíró mátrix, a rendszermátrix.
- $\mathbf{B}$ a szakaszra jellemző vektor, amely a beavatkozást határozza meg.

Az egyszerűség kedvéért foglalkozzunk SISO rendszerekkel, és korlátozzuk be, hogy az állapotváltozóink is egydimenziósak. Így y_k és x_k is egydimenziós, akárcsak u_k. Mindezek mellett adott egy költségfüggvény, amely meghatározza azt a hibát, amelyet az algoritmus minimalizál. Első példánkban célunk, hogy a rendszert egy kezdőállapotból egy másik, általunk előírt állapotba juttassuk. Így az optimalizációs feladat a következő alakot veszi fel:

$$\min \sum_{i=0}^{N} (\mathbf{x}_{k+i|k} - \mathbf{x}_{req})^2 \tag{3}$$

$$\mathbf{x}_{k+i+1|k} = \mathbf{A}\mathbf{x}_{k+i|k} + \mathbf{B}u_{k+i|k} \tag{4}$$

$$u_{down} < u_{k+i|k} < u_{up} \tag{5}$$

$$\mathbf{x}_{k+i|k} \in \mathbb{R}^n \tag{6}$$

$$u_{k+i|k} \in U \tag{7}$$

Ahol:

- N az időhorizont, eddig az időpontig nézzük a rendszer válaszait.
- $\mathbf{x}_{k+i|k}$ a k-edik időpontban a szakasz válaszai az időhorizonton i futó változóval.
- $u_{k+i|k}$ a k-edik időpontban a beavatkozó jel az időhorizonton i futó változóval.
- u_{down} a beavatkozó jel alsó határa.
- u_{up} a beavatkozó jel felső határa.
- $\mathbf{x}_{req}$ az általunk előírt állapotvektor.
- $U \in \{U_0, U_1, \ldots, U_9\}$

Az optimalizáció célja a (3) egyenlet minimalizálása, ez az object, míg a (4) és a (5) a betartandó határok. Triviális, hogy mind az u_k mind az x_k optimalizálandó változók. Mindezek mellet, a rendszer állapotvektorai folytonos értékeket kaphatnak, addig a beavatkozó jelek csak egy véges számossággal rendelkező halmazból kerülhetnek ki. Ezt az optimalizációs problémát Mixed Integer Linear Problem-nek (MILP) nevezzük.

2.2 A Várakozási Idő Határok (WTC)

Ahogyan a fentiekben is említettem, ez a szekció nem a járványügyi modellel foglalkozik, de itt lehet kitérni a várakozási határok (azaz waiting time constraints) kérdéskörére. A probléma alapja a következő: egy adott beavatkozást nem lehet hirtelen megváltoztatni, illetve nem célszerű túl hosszú ideig fenn tartani társadalmi és gazdasági okokból. Ezért létrehozunk olyan korlátozásokat, amelyek a beavatkozások hosszát szabályozzák, azaz meghatározzák, hogy a beavatkozó jel mennyi ideig lehet aktív. Fogalmazhatunk úgy is, hogy egy adott beavatkozó jelet akármikor ráköthetünk a rendszerre, azonban ha beavatkozunk, annak értékét egy minimális ideig tartanunk kell, és egy bizonyos idő után már nem alkalmazhatjuk ezt az értéket egészen addig, ameddig egy másik beavatkozást nem alkalmazunk. Ennek a megoldására újabb optimalizálandó változókat vezetünk be.

$$\mathbf{v_j} = [v_j^0, v_j^1, \ldots, v_j^n]^T$$

Ahol:

$$v_j^l = \begin{cases} 1, & \text{ha az adott } u \text{ aktív} \\ 0, & \text{egyébként} \end{cases}$$

Illetve feltesszük, hogy:

$$\sum_{l=0}^{n} v_j^l = 1 \tag{8}$$

Azaz $\mathbf{v_j}$ egy olyan vektor, amelynek minden eleme 0, kivéve azt az indexet, amelyien értékű u abban az adott időpillanatban. Például, ha $u = U_1$ és $U \in \{U_0, U_1, \ldots, U_9\}$, akkor $\mathbf{v_j} = [0, 1, 0, \ldots, 0]$.

Bevezetünk továbbá egy $\mathbf{H}$ számlálót is, amely számolja, hány darab v_j volt aktív az adott időpillanatig. Ha az előzőtől eltérő értékű v_j következik a szekvenciában, akkor nullázza magát. Mindezek mellett bevezetünk további két paramétert is, $\mathbf{W}_{Tmin}$-t és $\mathbf{W}_{Tmax}$-ot, amelyek meghatározzák, hogy az adott v_j index milyen hosszú ideig tartsa meg, és milyen hosszú idő után változtassa meg értékét. Működésüket a következőképpen lehet leírni:

$$\mathbf{H}_{j+1} = (\mathbf{H}_j + \mathbf{v}_j) \odot \mathbf{v}_j \tag{9}$$

Ahol a $\odot$ operátor az elemenkénti szorzást jelenti, azaz a Hadamard-szorzást.

$$\mathbf{v}_j^T \cdot (\mathbf{W}_{Tmin} - \mathbf{H}_{j+1}) \leq M \cdot (\mathbf{v}_j^T \cdot \mathbf{v}_{j+1}) \tag{10}$$

Ahol M egy nagy szám, amely az egyenlőtlenséget $\mathbf{v}_j = \mathbf{v}_{j+1}$ esetén biztosan tartja.

$$\mathbf{H}_{j+1} \leq \mathbf{W}_{Tmax} \tag{11}$$

Míg a (9) egyenlet a számlálásért felelős, addig (10) az alsó határ, a (11) felső határ előállításáért felelős.

2.3 Szimulációs eredmények

A következő szekcióban bemutatok néhány szimulációs eredményt Az optimalizációs problémát Python nyelven implementáltam a Pyomo és NumPy csomagok segítségével, a vizualizációt pedig a Matplotlib csomaggal oldottam meg. Ebben a példában a Gurobi solver-t használtam az optimalizációs probléma megoldása érdekében, mivel konvex problémával állunk szemben. (Konvex problémáról beszélünk, ha az optimalizálandó függvény konvex, az egyenlőségek affinok, az egyenlőtlenségek pedig konvexek.)

A rendszert leíró dinamikát meghatározó paraméterek:

$$\mathbf{A} = 1.1 \qquad \mathbf{B} = 1.0$$

Illetve az $\mathbf{U}$ halmaz , $\mathbf{W}_{Tmin}$ és $\mathbf{W}_{Tmax}$

$$\mathbf{U} \in \{-10, -9, -8 \ldots 10\} \qquad \mathbf{W}_{Tmin} = 2 \qquad \mathbf{W}_{Tmax} = 4$$

Legyen a $\mathbf{x}_0$ kiinduló és $\mathbf{x}_{req}$ referencia állapotok az alábbiak:

$$\mathbf{x}_0 = 0.0 \qquad \mathbf{x}_{req} = 100.$$

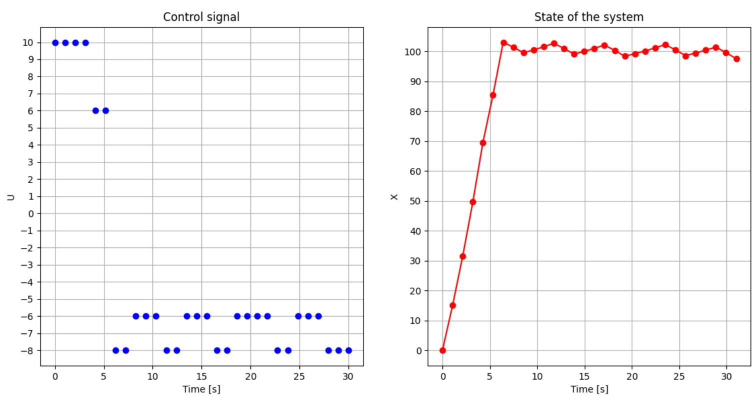

Figure 1: Balra a beavatkozó szekvencia, jobbra pedig a rendszer alakulása

Amint látható, a válasz körül fog oszcillálni a rendszer. Ennek oka, hogy a beavatkozójel nem illeszkedik a dinamikára, nem vesz fel olyan értéket, amellyel pontosan a referenciajelre irányítható a rendszer. Továbbá azt is megfigyelhetjük, hogy az oszcilláció meghatározott szakaszokból áll, ennek oka, hogy a beavatkozások idejét korlátoztuk. Megemlíthetjük, hogy ha nem lenne korlátozás a beavatkozójelben, akkor a rendszer oszcillációmentesen elérné a referenciajelet.

A példa teljessége érdekében kiterjesztettem a rendszert két dimenziós esetre is. Ebben az esetben a paraméterek az alábbiak voltak:

$$\mathbf{A} = \begin{pmatrix} 1 & 1.4 \\ 1.4 & 1. \end{pmatrix} \qquad \mathbf{B} = [1., 1.]$$

Mind $\mathbf{U}$ halmaz mind Mind az $\mathbf{U}$ halmaz, mind a $\mathbf{W}Tmin$, $\mathbf{W}Tmax$ paraméterek megegyeznek az előbbi egydimenziós esettel. Az időhorizont ebben az esetben, ahogyan az egydimenziós esetben is, 30 s volt.

A kiinduló $\mathbf{x}_0$ és a referenica $\mathbf{x}_{req}$ állapotok pedig az alábbiak:

$$\mathbf{x}_0 = [0., 0.] \qquad \mathbf{x}_{req} = [10., 10.]$$

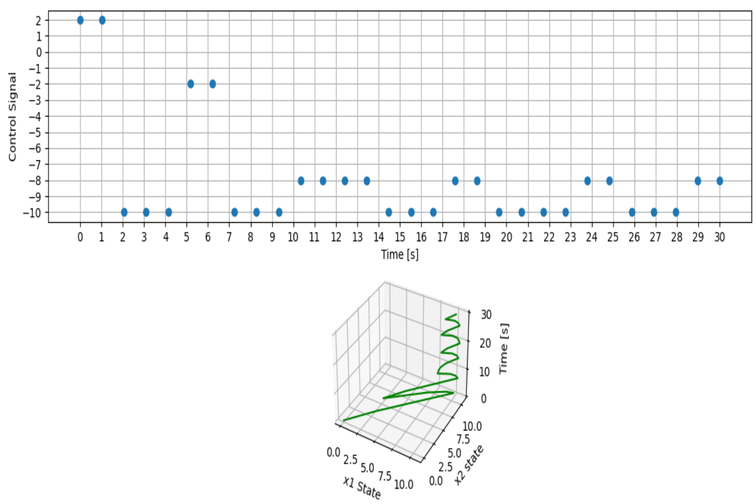

Figure 2: Fent a beavatkozó szekvencia, alul pedig a rendszer alakulása

3 Járványfolyamat modellje

3.1 A modell dinamikájának bemutatása

A következő szekcióban azt a járványmodellt fogjuk megnézni, amelyhez az előző fejezetben taglalt stratégiát építettem ki, apró módosításokkal. A folyamatmodell lényege a következő. A társadalmat különböző csoportokra bontjuk attól függően, hogy a fertőzés mely stádiumában vannak. Az átjárást az egyik halmazból a másikba differenciálegyenletek írják le, azaz ez a modell egy folytonos idejű modell. Lássuk az egyes csoportokat, illetve a dinamika leírását.

Adott a még meg nem fertőzött populáció (S), akik még nem kapták el a fertőzést, de potenciális betegek lehetnek. Adottak a látens (L) populáció, akik találkoztak már megfertőződtek, de még nem mutatnak semmilyen tünetet. Ezt követően a népesség a pre-szimptomatikus fertőző (P) csoportba kerül, akikből vagy a fertőzés tüneteit produkáló (I) halmazba vagy a fertőzés tüneteit nem produkáló halmazba (A) jutnak az egyedek. Azok, akik nem mutatják a fertőzés tüneteit, nagy valószínűséggel felépülnek a betegségből, ez megköveteli a felépült (R) halmaz definiálását. Azok, akik tüneteket mutatnak, vagy felépülnek és eljutnak az R halmazba, vagy kórházba kerülnek (H). Akik a kórházban vannak, vagy felépülnek, vagy meghalnak (D). A dinamikát a következő differenciálegyenletek fogják leírni:

$$S'(t) = -\beta \frac{[P(t) + I(t) + \delta A(t)]S(t)}{N} \tag{12}$$

$$L'(t) = \beta \frac{[P(t) + I(t) + \delta A(t)]S(t)}{N} - \alpha L(t) \tag{13}$$

$$P'(t) = \alpha L(t) - pP(t) \tag{14}$$

$$I'(t) = qpP(t) - \rho_1 I(t) \tag{15}$$

$$A'(t) = (1 - q)pP(t) - \rho_A A(t) \tag{16}$$

$$H'(t) = \rho_1 \nu I(t) - hH(t) \tag{17}$$

$$R'(t) = \rho_1(1 - \nu)I(t) + \rho_A A(t) + (1 - \mu)hH(t) \tag{18}$$

$$D'(t) = \mu hH(t) \tag{19}$$

Ahol:

- β: A fertőzési ráta.
- δ: A fertőzőképesség aránya az $A(t)$ csoportban.
- N: A teljes populáció mérete.
- α: Az $L(t)$ állapotból való átlépés aránya.
- p: A $P(t)$ állapotból való átlépés aránya.
- q: Annak a valószínűsége, hogy a $P(t)$ állapotból valaki az $I(t)$ állapotba lép.
- ρ_1: A fertőző egyének ($I(t)$) gyógyulási vagy átmeneti aránya.
- ρ_A: Az aszimptomatikus egyének ($A(t)$) gyógyulási vagy átmeneti aránya.
- ν: Annak a valószínűsége, hogy egy fertőző egyén ($I(t)$) kórházi kezelésre szorul.
- h: A kórházi betegek ($H(t)$) halálozási aránya.

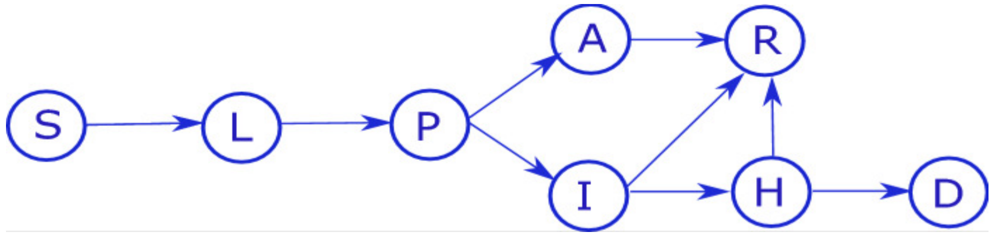

Figure 3: A populáció alakulása a járvány folyamán

A dinamikát az alábbi gráfábra mutatja be:

Mivel az optimalizáció során számunkra diszkrét idejű modell kell, ezért a fent említett differenciálegyenlet rendszert valamilyen numerikus módszerrel meg kell oldanunk. Ezt a megoldásom során negyed rendű Runge-Kutta módszerrel oldottam meg. Numerikus stabilitás érdekében normalizáltam az egész populáció számosságával az egyes állapotokat.

A modell kapcsán megállapíthatjuk, hogy a modell megmaradó tulajdonsággal bír, azaz ha összeadjuk az összes egyenletet, akkor a lakosság számosságával normálva pontosan egyet kapnánk. Azaz a modellben nem keletkeznek újabb egyének. Továbbá megállapíthatjuk, hogy a rendszerben, ha az állapotok normálva vannak:

$$\lim_{t\to\infty} |S(t) + R(t) + H(t)| = 1$$

Azaz mindenki vagy az S, vagy az R, vagy a D halmazba fog kerülni. A modell nem tér ki az újrafertőződés mechanikájára.

Kitérhetünk az optimalizációs problémára. Mint az előző fejezetben bemutattam, a probléma egy object-ből illetve annak határaiból áll.

$$min \sum_{i=0}^{N} u_i^2 + (S_i - S_{term})^2 \tag{20}$$

$$\mathbf{x}_{i+1} = F(\mathbf{x}_i) \tag{21}$$

$$\mathbf{x}_i[5] <= 20000 \tag{22}$$

Ahol:

- N az idő horizont.
- i a futó változó.
- u_i az i időpillanatban bevezetet intézkedés (beavatkozó jel).
- $\mathbf{x}_i$ az i időpillanatban a rendszer állapotvektorja.
- $\mathbf{x}_k[5]$ az i időpillantaban a kórházban lévő emberek száma.
- S_{term} a terminális állapota a rendszernek, az az S halmaz melyre a járvány már biztosan lecseng.

- S_i Az i. időpillanatban az S halmaz számossága

Az (20) egyenlet a pandémia céljának a megfogalmazását jelenti, azaz minimalizálni akarjuk a beavatkozások értékét, így csökkenteni a gazdasági károkat, illetve el akarjuk jutatni a rendszert egy olyan állapotba, ahol már a járvány lecseng ($\mathbf{x}_{term}$ terminális állapot). A (21) a rendszer dinamikájának mintavételezését jelenti, azaz az előző állapotból, hogyan jutunk a következőbe. A (22) korlátozásban pedig megszabjuk, hogy a kórházban lévő betegek száma egy értéket ne haladjon meg annak érdekében, hogy ne alakuljon ki kapacitás probléma a közegészségügyi rendszerben.

Ki lehet térni még a terminális halmaz vagy terminális állapot kérdéskörére. Terminális állapotnak nevezzük a járvány azon állapotát, amelyben nulla beavatkozás mellett is a kórházban ápoltak száma szigorúan monoton csökken, azaz kialakul a nyájimmunitás. Nyilvánvaló, hogy ha elérjük a terminális halmazt, akkor a járvány nem terjed tovább. Ilyen állapotokat kell elérnünk az irányítás során, ez a (20) kifejezés második tagjában nyilvánul meg, hasonlóan ahhoz, ahogyan az első példában el akartunk érni egy referencia állapotot. Azt megállapítani, hogy hol vannak ezek a terminális állapotok, már nagyon nehéz kérdés. Nem triviális egy nem lineáris modellnél megállapítani, hogy egy állapot elérhető-e vagy sem. Megoldásomban egy brute force megoldást használtam. Véletlenszerű kezdőállapotokból kiindítva figyeltem meg a rendszert, hogy azok az állapotok stabilak-e vagy sem. A véletlen inicializációt úgy valósítottam meg, hogy az egyes halmazok csak "ésszerű" értékeket vegyenek fel, és ne vegyenek fel irreális értékeket.

3.2 Szimulációs eredmények

A megoldást python nyelven implementáltam és szintén a pyomo, numpy könyvtárakat alkalmaztam, akárcsak az első fejezetben. Az optimalizációs probléma megoldásához Baron solvert használtam. A köveketkezőkben bemutatom a szimulációhoz használt paramétereket:

$$\mathbf{x}_{init} = [9800000 - 50, 50, 0., 0., 0., 0., 0., 0.]$$

Azaz a teljes népesség 9 800 000 fő míg azoknak a száma akik a látens halmazban vannak 50 fő. Továbbá az időre vonatkoztatott paraméterek:

$$t_{end} = 180 \qquad dt = 1$$

A teljes idő horizontotot 180 napnak vettem, és azt a megkötést alkalmaztam, hogy csak hetente hozhatunk rendelkezést, és azt a rendelkezést amelyet hoztunk egy hétig fenn is kell tartanunk. A mintavételezési idő 1 nap. A következő megfigyeléseket tehetjük meg. ha előírjuk a kórházban lévők számosságát akkor egy bizonyos meredekséggel, sebességgel tudjuk csak elérni a terminális halmazt, azaz időnek kell eltelnie ahhoz, hogy elérjük a nyájimmunitást. Azaz minnél nagyobb a kórházi kapacitás, annál gyorsabban túl tudunk esni a járványon. Azt is láthatjuk, kritikus az S halmaz, illetve a fertőzni képes halmazok (L,P,A,I) számossága, ezek egymáshoz képest viszonya dönti el, hogy nulla beavatkozás mellet, kialakulhat egy második hullám, egy újabb maximum a H halmazban.

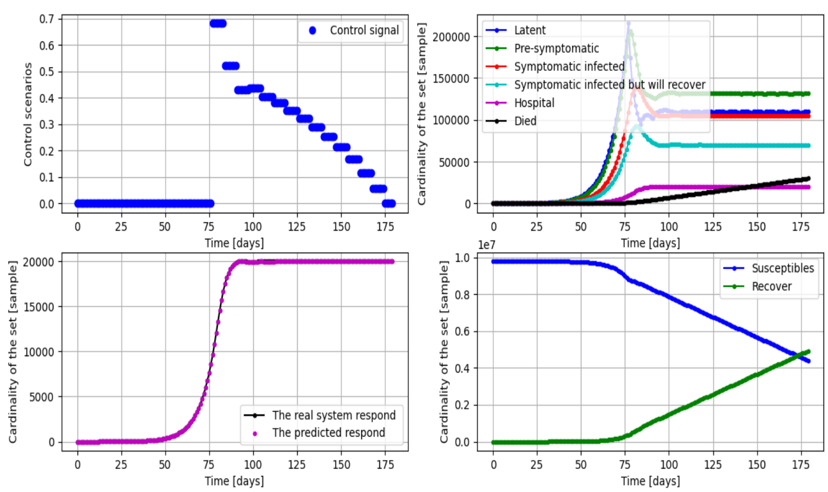

Figure 4: A rendszer változása az idő folyamán. Bal felső sarokban a beavatkozó jel szekvenciája. Jobb felső sarokban trainzens állapotok és az elhunytak száma látható. Jobb alsó sarokban a valódi (folytonos) és a prediktált (pontozott) rendszer a kórházban lévők számossága alapján. Jobb alsó sarokban a felépültek és még meg nem betegettek számossága