SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA CHEMICKEJ A POTRAVINÁRSKEJ TECHNOLÓGIE
Ústav informatizácie, automatizácie a matematiky

Pokročilé aspekty prediktívneho riadenia

Habilitačná práca

2022 **Martin Klaučo**

Poďakovanie

Habilitačná práca sumarizuje nielen aktivitu predkladateľa, ale aj celý kolektív, ktorý predkladateľa formoval počas jeho kariéry, a preto na prvom mieste vyjadrujem vďaku celému Oddeleniu informatizácie a riadenia procesov. Najväčšie poďakovanie však patrí prof. Ing. Michalovi Kvasnicovi, PhD., ktorý mi vytváral priestor na odborný aj osobnostný rast od prvého momentu našej spolupráce. V druhom rade patrí obrovské poďakovanie dlhoročnému vedúcemu oddelenia prof. Ing. Miroslavovi Fikarovi, DrSc., ktorý dbá aj vždy dbal na kvalitné vedecké výstupy. Do tretice sa chcem veľmi pekne poďakovať doc. Ing. Monike Bakošovej, CSc. za jej podporu a rady najmä v uplynulom období.

Ďalej sa chcem poďakovať mojím najbližším vedeckým spolupracovníkom doc. Ing. Jurajovi Oravcovi PhD., Ing. Martinovi Kalúzovi, PhD. a Ing. Jurajovi Holazovi, PhD., bez ktorých by vedecké práce s ktorými postupujem do habilitačného konania, nevznikli.

Martin Klaučo Bratislava, 2022

Mojim sestrám, Kláre a Dominike

Obsah

Kapitola 1

Úvod

Prediktívne riadenie (z angl. *Model Predictive Control* - MPC) patrí medzi hlavné piliere nielen vedeckého bádania, ale aj priamej implementácie riadiacich stratégii v priemyselných prevádzkach. Predkladaná habilitačná práca si dáva za cieľ priblížiť niektoré z pokročilých aspektov v prediktívnom riadení a jeho aplikácií. Táto práca predstaví najmä pokročilé koncepty v zostavovaní problému optimálneho riadenia vo forme MPC. Taktiež sa v tejto práci budeme zaoberať spôsobom výpočtu optimálneho akčného zásahu s cieľom urýchliť tento proces.

Z historického hľadiska boli prediktívne regulátory používané práve na riadenie a koordináciu jednoduchých, resp. nízko úrovňových riadiacich slučiek. Hlavnou úlohou koordinátora je voľba optimálnych žiadaných veličín pre nízko-úrovňové regulátory a nie priamy výpočet akčných zásahov. Prvý stavebný kameň v tomto prístupe k riadeniu bol publikovaný v roku 1979 [CR79], pod názvom "Dynamic Matrix Control", čo sa považuje za predchodcu klasického prediktívneho riadenia. Samozrejme, v posledných troch dekádach boli vyriešenie mnohé teoretické [MRRS00, May14] ako aj implementačné [QB03, WGBM03] problémy spojené s nasadením optimálnych regulátorov, čiže bolo možné nasadiť MPC aj priamo na riadenie komplexných procesov, bez nutnosti riešiť nízko-úrovňové riadenie. Avšak koncept viac úrovňového riadenia je v priemyselnej praxi veľmi žiadaný, čo demonštrujú aj nedávno publikované teoretické či aplikačné vedecké práce [LGKG22, LLR+19, HVHDG19]. Okrem toho, že výskum v oblasti nadradených regulátorov je aktuálny v rámci vedeckej aj priemyselnej praxe, najdôležitejším argument sa jeho ďalší rozvoj je schopnosť výrazného zvýšenia kvality riadenia bez nutnosti revitalizácie aktuálnych riadiacich slučiek.

Zostaviť prediktívny regulátor, ktorého cieľom je riadiť konkrétny systém patrí medzi tie "jednoduchšie" vedecké úlohy. Problémom v tejto časti je najmä pochopiť fyzikálne princípy dynamiky riadeného systému a samozrejme konkretizovať ciele riadenia. Medzi náročnejšie úlohy patrí aspekt riešenia týchto optimalizačných problémov. Práve kombinácia zostavenia prediktívnych regulátorov ako nadradených regulátorov má za následok,

že vzniknuté optimalizačné problémy sú veľkorozmerné, čo do počtu optimalizačných premenných ako aj počtu ohraničení. Práve z tohto pohľadu sa v práci budeme zaoberať aj možnosťami zrýchlenia získania optimálneho riešenia. Táto habilitačná práca si nekladie za cieľ fundamentálne predostrieť nové koncepty riešenia optimalizačných úloh, ale načrtne možnosti efektívneho využitia iných metód na istú triedu problémov.

MPC problémy môžeme riešiť dvoma spôsobmi, implicitne, resp., "online", čiže zostavený optimalizačný problém numericky vyriešime pre aktuálnu hodnotu počiatočnej podmienky, alebo explicitne, čiže "offline", kde je zákon riadenia skonštruovaný ako sústava lineárnych rovníc. Oba prístupy majú svoje výhody aj nevýhody, pričom oba prístupy boli aktívne využívané pri riešení problémov optimálneho riadenia v predkladanej práci. Medzi hlavné výzvy pri implementácií prediktívneho riadenia je však čas, za aký vieme hodnotu optimálneho akčného zásahu získať. Na strane implicitného riešenia si ukážeme, ako môžeme urýchliť výpočet pomocou vhodnej inicializácie optimalizačného problému [KKK19]. V tejto práci ukážeme, ako je možné využiť metódy strojového učenia na určenie počiatočného bodu pri numerickom riešení optimalizácie. Vzhľadom na to, že ide o aktívnu oblasť výskumu, v habilitačnej práci predstavíme aj nové výsledky, ktorých cieľom je prepojiť výhody prediktívneho riadenia s metódami strojového učenia, pomocou ktorých vieme získať zákony riadenia v explicitnom tvare [LKFM20, LK18].

V neposlednom rade spomeňme aj možnosti riešenia MPC pomocou parametrickej optimalizácie, teda skonštruovania zákona riadenia v tvare explicitne danej funkcie. Nezabudnime pripomenúť, že fundamentálny odkaz v rovine explicitného prediktívneho riadenia zanechali vedecké práce [BMDP02, BBM02]. Využitie explicitného prístupu ku konštrukcii zákonov riadenia bolo využité aj práve pri zostavení nadradených prediktívnych regulátorov. Práve v kombinácií s požiadavkou veľmi rýchleho vyhodnotenia optimálneho akčného zásahu sme MPC v tvare supervízora navrhovali pri riadení koordinácie nízko-úrovňového riadenia pH v zmiešavači kvapalín [HKD+18]. Keďže ale zložitosť explicitného MPC rastie dramaticky s rozmermi riadeného systému, tak v tomto prípade sme zvolili prístup tzv. bez-regiónového MPC [DJKK16, TSG+20, DKJK17], ktorého pamäťové požiadavky sú rádovo nižšie v porovnaní s tradičným prístupom ku konštrukcii explicitného prediktívneho regulátora.

1.1 Prínosy práce

1.1.1 Syntéza pokročilých prediktívnych regulátorov

Celkový prínos k formulácií problémov spojených s nadradeným riadením bol sumarizovaný v monografii [KK19b]. V tejto monografii sú sumarizované tri základné prístupy k návrhu nadradených regulátorov:

1. riadenie slučiek s PID regulátormi,

2. riadenie slučiek s relé (on/off) riadením,

3. riadenie slučiek s nízko-úrovňovým MPC.

Vyššie uvedená monografia rozširuje základné koncepty formulácie MPC problémov, ktoré vznikli z priložených vedeckých prác [HKD+18, KK17, KKK17], ako aj ostatnej práce autora [HKK17, DKK15, KKK15].

Pri návrhu prediktívneho riadenia ako supervízora sme sa sústredili na tri hlavné časti

1. formulácia agregovanej účelovej funkcie, ktorá zohľadní ciele riadenia nielen z pohľadu nadradeného regulátora, ale aj kvalitatívne parametre nižšej riadiacej vrstvy,

2. rozšírenie modelu o ohraničenia charakterizujúce vnútorné riadiace obvody,

3. efektivitu riešenie optimalizačného problému, najme vzhľadom na nadradené riadenie vnútorných slučiek s MPC regulátormi.

Aj keď je téma nadradených regulátorov rozoberaná v mnohých vedeckých prácach, ako napríklad v [GNN18, GK02], prezentovaný prístup k syntéze nadradených prediktívnych regulátorov sa líši v dvoch kľúčových oblastiach:

1. prezentuje systematický prístup k syntéze regulátora vzhľadom na dynamiku vnútornej vrstvy a všetky technologické ohraničenia,

2. autorom navrhované MPC sa nespráva ako filter žiadanej veličiny, ale je to autonómny regulátor, ktorý môže fungovať aj ako "bias corrector" v prípade, že nastane konštantná porucha na riadenom procese.

Druhým systematickým prínosom v prípade syntéze pokročilých MPC regulátorov je nový prístup k aspektu priebežnej (online) laditeľnosti explicitných regulátorov. Je známe, že pri konštrukcii explicitného MPC regulátora musíme zafixovať štruktúru, teda model riadeného systému, predikčný horizont a váhové matice účelovej funkcii. Ukážeme, akým spôsobom môžeme naformulovať úlohu prediktívneho riadenia s variabilnými váhovými maticami, s tým, že vieme zostrojiť explicitný regulátor, ktorý bude spĺňať podmienky garancií stability a rekurzívnej riešiteľnosti [OK22].

1.1.2 Metódy strojového učenia v prediktívnom riadení

Hlavným prínosom autora v oblasti zrýchlenia výpočtových časov MPC stratégií je využitie metód strojového učenia na zefektívnenie inicializácie metódy aktívnych množín (z angl. Active Set Method - ASM). Vo vedeckej práci [KKK19] rozoberáme spôsob skonštruovania prediktora pomocou metód strojového učenia, ktorý na základe stavového merania odhadne, ktoré ohraničenia by mali byť aktívne, a indexmi týchto ohraničení

inicializujeme metódu aktívnych množín. Vzhľadom na skoro-optimálnu inicializáciu ASM, sa výpočet optimálneho akčného zásahu zredukuje na

1. buď na numerické riešenie kvadratického problému Newtonov metódou, ak "trafíme" optimálnu inicializáciu aktívnych ohraničení,

2. alebo na výraznú redukciu nutnosti vykonať väčšie množstvo iterácií, keďže väčšina aktívnych ohraničení bude odhadnutá dopredu.

Najväčšiu výhodou tohto prístupu je garancia, že problém optimálneho riadenia bude vždy vyriešený optimálne, pretože numerickú metódu aktívnych množím nepreskočíme, len ju inicializujeme.

Okrem vyššie spomenutého prieniku strojového učenia do prediktívneho riadenia, predostrieme nový koncept konštrukcie explicitných regulátorov vo forme neurónových sietí. Aj keď výsledky v tejto oblasti nepatria medzi kľúčové piliere tejto práce, patria medzi aktuálne témy, ktoré autor práce rieši [KK19a, LKFM20, LKKM19].

Kapitola 2

Syntéza pokročilých prediktívnych regulátorov

2.1 Základný koncept prediktívnych regulátorov

Formulácia pokročilých prediktívnych regulátorov vychádza zo základných formulácií stavovej regulácie, alebo riadenia na žiadanú hodnotu. Základným stavebným kameňom matematickej formulácie je minimalizácia kvadratickej účelovej funkcie, vzhľadom na predikciu správania sa lineárneho modelu riadeného systému, pričom uvažujeme dodatočné ohraničenia v tvare nerovnosti pre ako tak pre stavové premenné, tak aj pre manipulované veličiny. Zápis takého optimalizačného problému vyzerá nasledovne

$$\min_{U} \; x_N^\intercal \mathcal{Q}_\mathrm{N} x_N + \sum_{k=0}^{N-1} \left(x_k^\intercal Q_\mathrm{N} x_k + u_k^\intercal Q_\mathrm{u} u_k\right) \tag{2.1a}$$

$$\text{v.n.} \quad x_{k+1} = Ax_k + Bu_k, \qquad k \in \mathbb{N}_0^{N-1}, \tag{2.1b}$$

$$u_k \in \mathcal{U}, \qquad k \in \mathbb{N}_0^{N-1}, \tag{2.1c}$$

$$x_k \in \mathcal{X}, \qquad k \in \mathbb{N}_0^{N-1}, \tag{2.1d}$$

$$x_N \in \mathcal{X}_\mathrm{N}, \tag{2.1e}$$

$$x_0 = x(t), \tag{2.1f}$$

kde veličina N prestavuje predikčný horizont, matice $Q_\mathrm{N} \in \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{x}}$, $Q_\mathrm{N} \in \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{x}}$ a $Q_\mathrm{u} \in \mathbb{R}^{n_\mathrm{u} \times n_\mathrm{u}}$ reprezentujú váhové parametre. Ďalej uvažujeme dynamický diskrétny matematický model riadeného systému (2.1b), pričom platí, že $A \in \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{x}}$, $B \in \mathbb{R}^{n_\mathrm{u} \times n_\mathrm{x}}$, a zároveň, že riadený systém je riaditeľný. Každé z ohraničení je (2.1b)-(2.1d) je reprodukované pre každý moment predikcie, teda $k = \{0, 1, \ldots, N-1\}$ resp. $k \in \mathbb{N}_0^{N-1}$. Pre každú veličinu stavového modelu uvažujeme polyhedrálne ohraničenia symbolizované množinami $\mathcal{U}$, $\mathcal{X}$, pričom finálny stav je obmedzený terminálnou množinou $\mathcal{X}_\mathrm{N}$ a penalizovaný terminálnou penalizáciou $\mathcal{Q}_\mathrm{N}$. Začiatočnou podmienkou na riešenie optimalizačného

problému je stavové meranie reprezentované veličinou $x(t)$ v rovnici (2.1f). Výsledkom riešenia optimalizačného problému je sekvencia predikovaných optimálnych akčných zásahov združená vo vektore

$$U = \left[u_0^\intercal, \quad \ldots, \quad u_{N-1}^\intercal\right]^\intercal. \tag{2.2}$$

Problém zostavený v rovnici (2.1) prestavuje MPC regulátor pre stavovú reguláciu a jeho implementácia v režime posuvného horizontu (z angl. *receding horizon policy*) stabilizuje riadený systém a garantuje rekurzívnu riešiteľnosť [MRRS00]. Pre mnohé aplikácie je však táto formulácia nevýhodná, pretože neuvažuje výstupnú rovnicu, ako aj neriadi systém na žiadanú hodnotu. Preto v uplynulom období vznikli mnohé variácie formulácie z rovnice (2.1), ktoré sú rozšírené o penalizáciu regulačnej odchýlky namiesto stavov, resp. rozšírené o integračnú činnosť, alebo rozšírené o modelovanie porúch [PR03]. Samozrejme môžeme hovoriť aj o formách nelineárneho MPC riadenia, kedy predikčným modelom je forma diskretizovanej diferenciálnej rovnice, ktorá vernejšie opisuje správanie sa riadeného systému. Tu však narážame na očividné problémy s riešením takýchto foriem nelineárnych optimalizačných problémov.

Problém v rovniciach (2.1a)-(2.1f) patrí do triedy kvadratických optimalizačných problémov, ktoré vieme riešiť numericky pomocou mnohých metód, ako sú

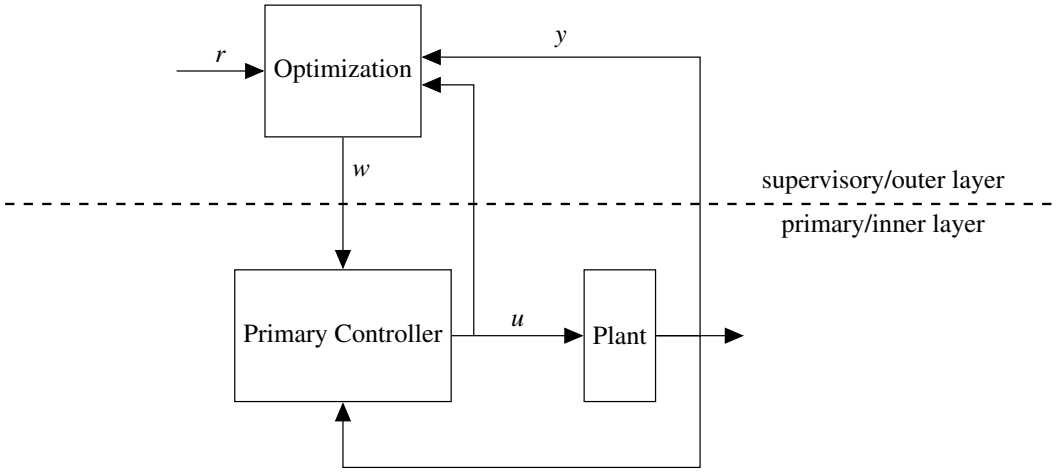
- metódy aktívnych množín (Active Set Method - ASM) [WN10, FBD08],
- metódy vnútorných bodov (Interior Point Methods) [NW06],
- rôzne formy gradientových metód [RJM12, KF11, PB14],

alebo parametricky s cieľom získať explicitnú formu zákona riadenia [BMDP02, GBN11]. Ak budeme predpokladať lineárnu dynamiku v dynamickom modeli, ako aj polyhedrálne ohraničenia na premenné, tak potom aj rozšírené formulácie MPC vedú v konečnom dôsledku na formuláciu kvadratického programovania (alebo lineárneho, ak budeme uvažovať jedno alebo nekonečno normy v účelovej funkcií). Z tohto dôvodu sa budeme aj v kapitole 3 venovať práve tejto triede problémov.

Táto sekcia zadefinovala základný matematický priestor pre formuláciu pokročilých MPC stratégií, a nasledujúca podkapitola bližšie priblíži koncept nadradených regulátorov.

2.2 Syntéza nadradených MPC stratégií

Zo základnej MPC formulácie (2.1) ako aj rôznych rozšírení budeme vychádzať pri syntéze pokročilých foriem MPC regulátora. V prvom priblížení si predstavme všeobecnú formuláciu prediktívneho regulátora, ktorého úlohou je koordinovať nízko úrovňové regulátory (z angl. *MPC-based Reference Governor - MPC-RG*). Výsledky z tejto sekcie je možné nájsť v prácach autora [KK19b, HKD+18, KK17].



Obr. 2.1: Základná schéma prediktívneho regulátora v polohe nadradeného riadiaceho člena.

Pre lepšie pochopenie zvolenej notácie premenných sa pozrime na schému na Obr. 2.1, kde môžeme vidieť celkové zapojenie koordinátora v tvare MPC regulátora spolu s nižšou úrovňou riadenia. V tejto schéme vystupuje premenná r ako užívateľsky zadefinovaná žiadaná hodnota, w ako optimalizovaná žiadaná hodnota pre nízko-úrovňový regulátor, premenná u ako akčný zásah a premenná y ako meraný výstup.

Uvažujme formuláciu:

$$\min_{W} \ \ell_{\mathrm{N}}(x_N) + \sum_{k=0}^{N-1} \ell\left(x_k, y_k, u_k, w_k\right) \tag{2.3a}$$

$$\text{v.n.} \quad x_{k+1} = f(x_k, u_k), \qquad k \in \mathbb{N}_0^{N-1}, \tag{2.3b}$$

$$u_k = h(x_k, w_k), \qquad k \in \mathbb{N}_0^{N-1}, \tag{2.3c}$$

$$y_k = g(x_k, u_k), \qquad k \in \mathbb{N}_0^{N-1}, \tag{2.3d}$$

$$u_k \in \mathcal{U}, \qquad k \in \mathbb{N}_0^{N-1}, \tag{2.3e}$$

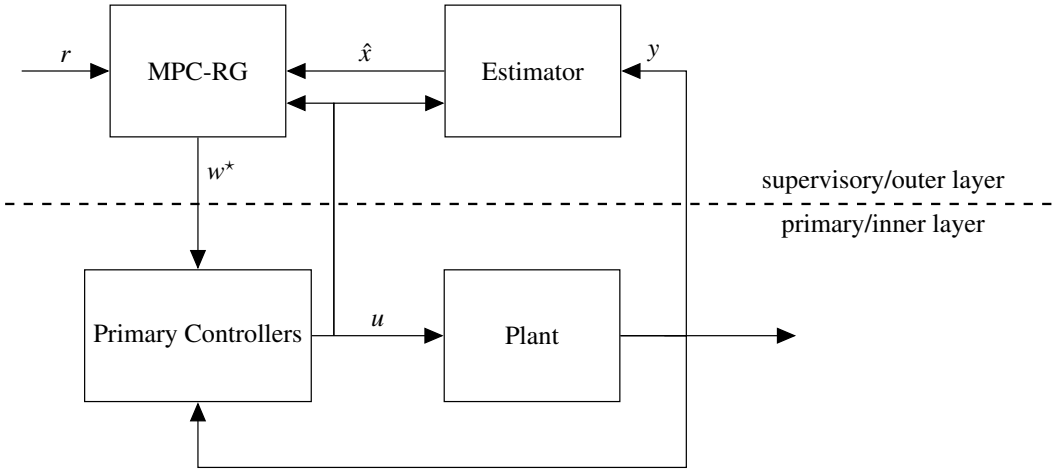
$$x_k \in \mathcal{X}, \qquad k \in \mathbb{N}_0^{N-1}, \tag{2.3f}$$

$$y_k \in \mathcal{Y}, \qquad k \in \mathbb{N}_0^{N-1}, \tag{2.3g}$$

$$w_k \in \mathcal{W}, \qquad k \in \mathbb{N}_0^{N-1}, \tag{2.3h}$$

$$x_0 = x(t), \tag{2.3i}$$

kde optimalizovanou premennou je $W = \left[w_0^\intercal, \ \ldots, \ w_{N-1}^\intercal\right]^\intercal$, čo reprezentuje žiadanú veličinu pre nižšiu úroveň riadenia. Rovnice (2.3b) a (2.3d) reprezentujú dynamiku riadeného systému, pričom rovnica (2.3c) je zákon riadenia vyjadrujúci správanie nízko-úrovňového riadenia. Oproti štandardnej formulácii MPC (2.1) uvažujeme ohraničenia v tvare nerovnosti aj pre výstupné veličiny (2.3g), ako aj pre ohraničenie optimalizovanej



Obr. 2.2: Schéma regulačného obvodu s pozorovateľom stavov.

žiadanej hodnoty (2.3h).

Počiatočnú podmienku uvádzame len pre stavy, ale v bežnej praxi sa do výrazu (2.3i) zakomponujú aj aktuálne hodnoty referencie r_k alebo aktuálne hodnoty porúch v prípade použitia prístupu modelovania porúch. Vzhľadom na tento fakt uvádzame aj schému zapojenia MPC-RG stratégie aj spolu s pozorovateľom stavov a porúch, keďže málokedy sa stane, že meranie stavov systému je priamo dostupné. Schéme na Obr. 2.2 zobrazuje práve prístup k takémuto zapojenie. Aj keď v schéme MPC-RG vystupuje celá vnútorná slučka ako riadený systém, v tomto prípade si je dôležité uvedomiť, že pozorovateľ stavov má za úlohu rekonštruovať len hodnotu stavov riadeného procesu a poprípade hodnotu nemerateľných porúch.

2.2.1 Formulácia MPC pre koordináciu nízkoúrovňových PID slučiek

Uvažujme, že celkový model PID regulátorov v stavovom opise sa dá napísať ako

$$x_{\mathrm{r},i}(t+T_\mathrm{s}) = A_{\mathrm{r},i}x_{\mathrm{r},i}(t) + B_{\mathrm{r},i}e_i(t), \quad i \in \mathbb{N}_1^p, \tag{2.4a}$$

$$u_i(t) = C_{\mathrm{r},i}x(t) + D_{\mathrm{r},i}e_i(t), \quad i \in \mathbb{N}_1^p, \tag{2.4b}$$

kde p definuje celkový počet PID regulátorov, $e_i(t)$ predstavuje regulačnú odchýlku pre i-tu regulačnú slučku a $x_{\mathrm{r},i}$ reprezentuje vektor interných stavov i-teho PID regulátora. Jednoduchými skladaním čiastkových matíc z rovnice (2.4) dostaneme agregovaný matematický model nízko-úrovňového PID riadenia

$$x_\mathrm{r}(t+T_\mathrm{s}) = A_\mathrm{r}x_\mathrm{r}(t) + B_\mathrm{r}e(t), \tag{2.5a}$$

$$u(t) = C_\mathrm{r}x_\mathrm{r}(t) + D_\mathrm{r}e(t). \tag{2.5b}$$

Pre potreby odvodenia kompletného modelu pre uzavretý regulačný obvod pre vnútornú vrstvu riadenia (viď Obr. 2.2) si definujeme pomocný stavový vektor $\widetilde{x}$ ako

$$\widetilde{x} = \begin{bmatrix} x_{\mathrm{r}} \\ x \end{bmatrix}. \tag{2.6}$$

Potom vieme napísať model systému "regulátor-proces" v otvorenej slučke pomocou nasledovne

$$\widetilde{x}(t+T_{\mathrm{s}}) = A_{\mathrm{OL}}\widetilde{x}(t) + B_{\mathrm{OL}}e(t), \tag{2.7a}$$
$$u(t) = C_{\mathrm{OL,u}}\widetilde{x}(t) + D_{\mathrm{OL,u}}e(t), \tag{2.7b}$$
$$y(t) = C_{\mathrm{OL,y}}\widetilde{x}(t) + D_{\mathrm{OL,y}}e(t), \tag{2.7c}$$

kde,

$$A_{\mathrm{OL}} = \begin{bmatrix} A_{\mathrm{r}} & 0 \\ BC_{\mathrm{r}} & A \end{bmatrix}, \tag{2.8a}$$
$$B_{\mathrm{OL}} = \begin{bmatrix} B_{\mathrm{r}} \\ BD_{\mathrm{r}} \end{bmatrix}, \tag{2.8b}$$
$$C_{\mathrm{OL,u}} = \begin{bmatrix} C_{\mathrm{r}} & 0 \end{bmatrix}, \tag{2.8c}$$
$$D_{\mathrm{OL,u}} = D_{\mathrm{r}}, \tag{2.8d}$$
$$C_{\mathrm{OL,y}} = \begin{bmatrix} DC_{\mathrm{r}} & C \end{bmatrix}, \tag{2.8e}$$
$$D_{\mathrm{OL,y}} = DD_{\mathrm{r}}. \tag{2.8f}$$

Pričom, ak uvažujeme štandardnú zápornú spätnú väzbu, kde regulačná odchýlka $e(t)$ je daná ako

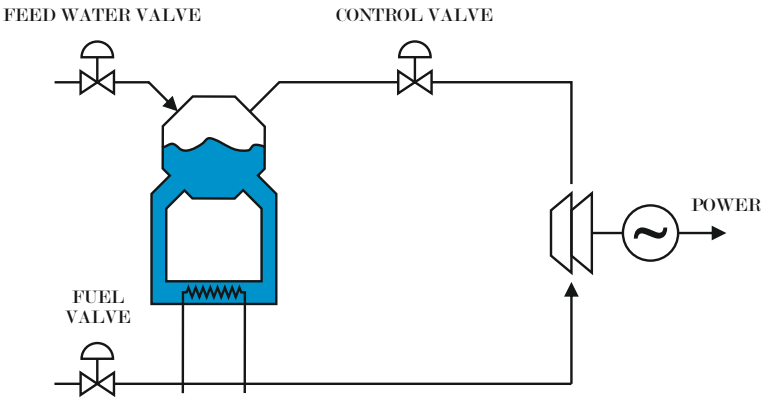
$$e(t) = w(t) - y(t), \tag{2.9}$$

tak potom platí

$$y(t) = (I + D_{\mathrm{OL,y}})^{-1} C_{\mathrm{OL,y}}\widetilde{x}(t) + (I + D_{\mathrm{OL,y}})^{-1} D_{\mathrm{OL,y}}w(t). \tag{2.10}$$

Ak skombinujeme rovnice (2.9) a (2.10) spolu s (2.7) tak dostaneme finálny model uzavretého regulačného obvodu

$$\widetilde{x}(t+T_{\mathrm{s}}) = A_{\mathrm{CL}}\widetilde{x}(t) + B_{\mathrm{CL}}w(t), \tag{2.11a}$$
$$u(t) = C_{\mathrm{CL,u}}\widetilde{x}(t) + D_{\mathrm{CL,u}}w(t), \tag{2.11b}$$
$$y(t) = C_{\mathrm{CL,y}}\widetilde{x}(t) + D_{\mathrm{CL,y}}w(t). \tag{2.11c}$$



Obr. 2.3: Schéma výroby elektrickej energie pomocou parnej turbíny [ÅB00, KK19b].

Pričom jednotlivé matice v (2.11) sú vyjadrené pomocou

$$A_{\mathrm{CL}} = A_{\mathrm{OL}} - B_{\mathrm{OL}} \left(I + D_{\mathrm{OL,y}}\right)^{-1} C_{\mathrm{OL,y}}, \tag{2.12a}$$

$$B_{\mathrm{CL}} = B_{\mathrm{OL}} - B_{\mathrm{OL}} \left(I + D_{\mathrm{OL,y}}\right)^{-1} D_{\mathrm{OL,y}}, \tag{2.12b}$$

$$C_{\mathrm{CL,u}} = C_{\mathrm{OL,u}} - D_{\mathrm{OL,u}} \left(I + D_{\mathrm{OL,y}}\right)^{-1} C_{\mathrm{OL,u}}, \tag{2.12c}$$

$$D_{\mathrm{CL,u}} = D_{\mathrm{OL,u}} - D_{\mathrm{OL,u}} \left(I + D_{\mathrm{OL,y}}\right)^{-1} D_{\mathrm{OL,u}}, \tag{2.12d}$$

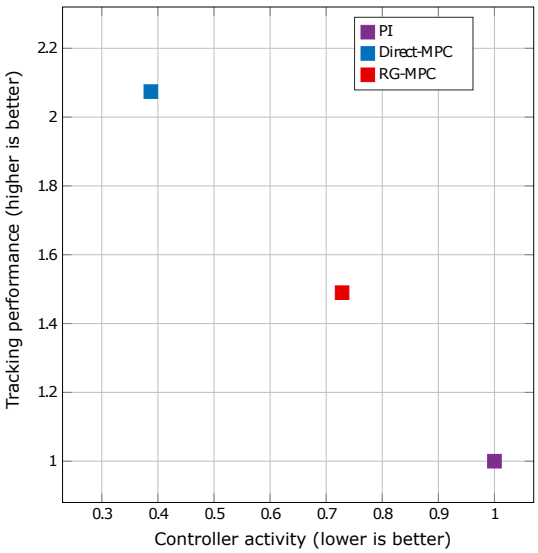
$$C_{\mathrm{CL,y}} = \left(I + D_{\mathrm{OL,y}}\right)^{-1} C_{\mathrm{OL,y}}, \tag{2.12e}$$

$$D_{\mathrm{CL,y}} = \left(I + D_{\mathrm{OL,y}}\right)^{-1} D_{\mathrm{OL,y}}. \tag{2.12f}$$

Keďže máme zostrojený model pre nízko-úrovňový regulátor ako aj pre riadený proces, tak môžeme zostaviť matematickú formuláciu pre MPC-RG stratégiu. Rovnice (2.11a) až (2.11c) zodpovedajú predikčnému modelu v optimalizačnom probléme (2.3b) až (2.3d). Keďže riadenie pomocou PID regulátorov je spojité, ako aj predpokladaný model riadeného procesu je stavový opis, tak potom celková formulácia MPC-RG optimalizačného problému sa v konečnom dôsledku zredukuje na úlohu kvadratického programovania.

Hlavné výsledky v rámci návrhu MPC-RG stratégie pre koordináciu PID regulátorov zobrazujeme na Obr. 2.4. Toto porovnanie vzišlo zo simulačného scenára pri riadení výroby elektrickej energie pomocou parnej turbíny [ÅB00, ÅE72], ktorej schematické zobrazenie predstavujeme na Obr. 2.3. V tomto porovnaní sa pozeráme na tri scenáre riadenia. Prvý scenár uvažuje, že systém riadime len pomocou skladby PID regulátorov. V druhom scenári použijeme výlučne prediktívny regulátor na riadenie parnej turbíny a v treťom scenári implementujeme MPC-RG stratégiu na základe odvodeného modelu v rovnice (2.11). Výsledky sú bližšie vysvetlené a odvodené v [KK17].

Z porovnania je zrejmé, že ak nasadíme MPC supervízora, tak na úrovni vstupov vieme ušetriť skoro 30% zdrojov, pričom na úrovni dosiahnutia referenčnej hladiny zvýšime kvalitatívny parameter o skoro 50%. Pre porovnanie uvádzame aj využitie priameho



Obr. 2.4: Možnosti zvýšenia kvality riadenia nasadením MPC koordinátora, pričom všetky hodnoty sú znormalizované vzhľadom na kvalitatívne vyhodnotenie riadenia pomocou skladby PID regulátorov.

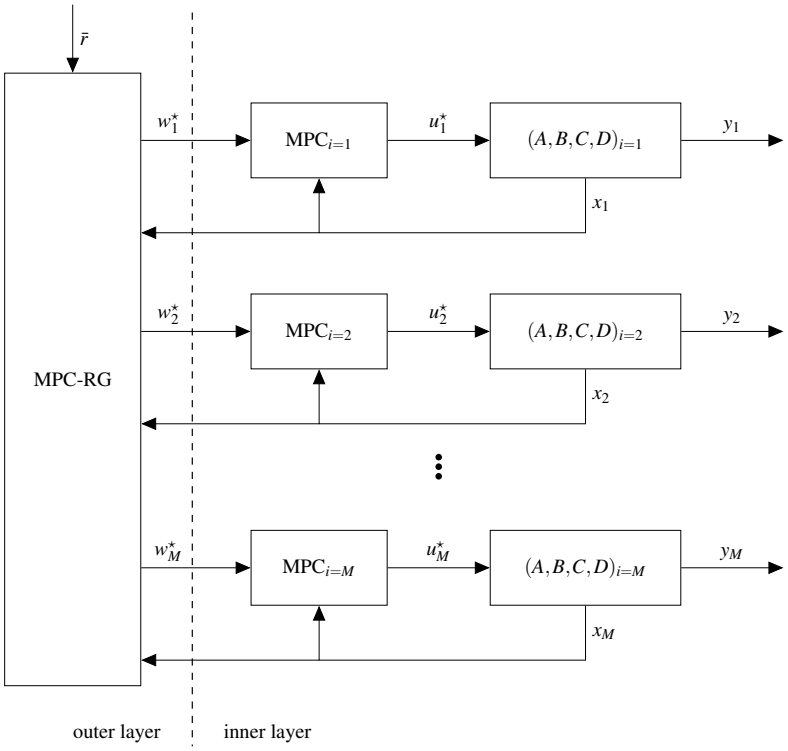
riadenia pomocou MPC, ktoré z pochopiteľných dôvodov výrazne vyhráva v kvalitatívnych aspektoch nad PID riadením resp. aj nad MPC-RG stratégiou.

2.2.2 Formulácia MPC pre koordináciu nízkoúrovňového optimálneho riadenia

Ako sme už naznačili v úvodnej kapitole, využitie MPC sa postupom času dostalo aj na nižšiu úroveň, t.j. prediktívne regulátory sa stali priamymi riadiacimi členmi procesov. Avšak aj v týchto prípadoch, sa častokrát stane, že je potrebné ich výkon koordinovať. Najmä z toho dôvodu, že nízkoúrovňové regulátory majú za hlavný cieľ riadiť a stabilizovať daný proces, a ich úlohou nie je komunikovať alebo uvažovať situáciu v inej časti technologickej prevádzky. Túto úlohu spĺňa práve koordinačný prediktívny regulátor, ktorý cez modifikáciu žiadanej veličiny pre nižšie úrovne riadenia zabezpečí zvýšenie kvalitatívnych parametrov. Výsledky z tejto sekcie je možné nájsť v prácach autora [KK19b, HKK17].

Schematické zapojenie MPC-RG stratégie pre koordináciu viacerých nízkoúrovňových MPC regulátorov znázorňujeme na Obr. 2.5. Zároveň formálne definujeme počet vnútorných MPC členov pomocou premennej M a predpokladáme, že každý z riedených procesov sa dá opísať pozorovateľným a riaditeľným lineárnym stavovým opisom.

Pre potreby odvodenia celkovej matematickej formulácie pre MPC-RG stratégiu v tomto prípade, uvažujme, že štandardnú formuláciu MPC z rovnice (2.1) zapíšeme



Obr. 2.5: Schéma zapojenia viacúrovňového prediktívneho riadenia pre koordináciu nízkoúrovňového MPC riadenia.

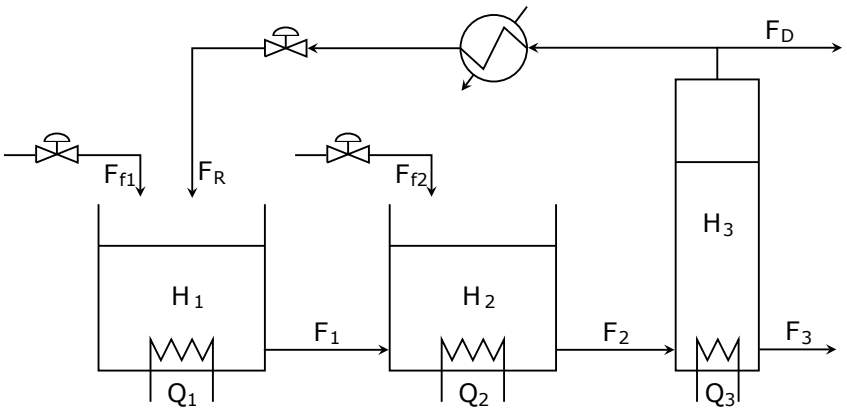
nasledovne

$$U_i^\star(t) = \text{MPC}_i(\theta_i(t)), \tag{2.13}$$

kde vektorom $\theta_i(t)$ rozumieme všetky vstupné parametre pre $i$ty MPC regulátor, teda napríklad

$$\theta = [x(t)^\intercal \; w(t)^\intercal]^\intercal, \tag{2.14}$$

kde $w(t)$ predstavuje žiadanú hodnotu, ktorú nastaví nadradený MPC regulátor. Zároveň platí, že $U_i^\star(t)$ je optimálne riešenie vnútorného MPC problému pozdĺž celého predikčného horizontu. Potom, celková zjednodušená formulácia nadradeného MPC koordinátora bude



Obr. 2.6: Technologický proces s dvoma reaktormi a odparovačom [SVR+10].

vyzerať nasledovne

$$\min_{W_{1,\dots,M},U_{1,\dots,M}} \sum_{j=0}^{\overline{N}} \sum_{i=1}^{M} \ell_{\mathrm{M},i}\left(r_{i,j}, y_{i,j}, w_{i,j}\right), \tag{2.15a}$$

$$\text{v.n.} \quad U^{\star}_{i,j}(\theta_{i,j}) = \mathrm{MPC}_i(\theta_{i,j}), \quad i \in \mathbb{N}_1^M, j \in \mathbb{N}_0^{\overline{N}}, \tag{2.15b}$$

$$u^{\star}_{i,j} = \Phi U^{\star}_{i,j}(\theta_{i,j}), \quad i \in \mathbb{N}_1^M, j \in \mathbb{N}_0^{\overline{N}}, \tag{2.15c}$$

$$x_{i,j+1} = A_i x_{i,j} + B_i u^{\star}_{i,j}, \quad i \in \mathbb{N}_1^M, j \in \mathbb{N}_0^{\overline{N}}, \tag{2.15d}$$

$$y_{i,j} = C_i x_{i,j} + D_i u^{\star}_{i,j}, \quad i \in \mathbb{N}_1^M, j \in \mathbb{N}_0^{\overline{N}}, \tag{2.15e}$$

$$\bar{x}_j \in \overline{\mathcal{X}}, j \in \mathbb{N}_0^{\overline{N}}, \tag{2.15f}$$

$$\bar{u}^{\star}_j \in \overline{\mathcal{U}}, j \in \mathbb{N}_0^{\overline{N}}, \tag{2.15g}$$

$$\bar{y}_j \in \overline{\mathcal{Y}}, j \in \mathbb{N}_0^{\overline{N}}. \tag{2.15h}$$

Formulácia v rovnici (2.15) uvažuje M kvadratických účelových funkcií reprezentujúcich partikulárne príspevky do celkového optimalizovaného kritéria jednotlivými MPC regulátormi a $\overline{N}$ reprezentuje predikčný horizont nadradeného MPC riadenia. Účelová funkcia teda môže vyzerať nasledovne:

$$\ell_{\mathrm{M},i}\left(r_{i,j}, y_{i,j}, w_{i,j}\right) = \|Q_{\mathrm{y},i}\left(r_{i,j} - y_{i,j}\right)\|_2 + \|Q_{\mathrm{w},j}\left(w_{i,j} - w_{i,j-1}\right)\|_2. \tag{2.16}$$

Rovnice (2.15b) a (2.15c) sú modelmi zákona riadenia, pričom tá druhá z nich reprezentuje extrakciu prvého akčného zásahu zo sekvencie $U^{\star}_{i,j}$ (teda zabezpečuje implementáciu metódy posuvného horizontu). Rovnice (2.15d) a (2.15e) s=u rovnicami modelov riadených systémov a množiny $\overline{\mathcal{X}}$, $\overline{\mathcal{U}}$, $\overline{\mathcal{Y}}$ predstavujú technologické ohraničenia na jednotlivé veličiny v riadených systémoch.

Vzhľadom na fakt, že rovnica regulátora v (2.15b) predstavuje optimalizačný problém sám o sebe, tak celková formulácia, tak ako je zostavená v (2.15) predstavuje *viacúrovňový optimalizačný problém*, ktorý nie je triviálne riešiť. Výsledky prezentované v monografii [KK19b] prezentujú dva hlavné prístupy, ako sa vysporiadať s riešením takéhoto optimalizačného problému. Zároveň je v tejto publikácií preberaná aj prípadová štúdia s riadením dvoch reaktorov s odparovačom, kde tri individuálne MPC regulátory zabezpečujú riadenie jednotlivých reaktorov a odparovača (schéma na Obr. 2.6), ktoré následne koordinujeme pomocou nadradenej MPC stratégie. Spomínané dva návrhy riešenia dvojúrovňového optimalizačného problému sa opierajú v prvom rade o explicitné MPC a v druhom rade o reformuláciu vnútorného problému pomocou Karush-Kuhn-Tuckerových (KKT) podmienok.

V prvom prípade sa spoliehame na to, že pre vnútorné MPC sa dá skonštruovať explicitné riešenie a takto dostaneme analytický tvar zákona riadenia v rovnici (2.15b). Aby sme vedeli povedať, ktorý región je v explicitnom riešení aktívny, teda, aké je hodnota akčného zásahu v rovnici (2.15c), priradíme každému regiónu binárnu premennú, ktorej hodnota nadobudne 1 v prípade, že akčný zásah spadá do tohto regiónu. Výsledný optimalizačný problém má následne charakter zmiešaného celočíselného programovania (z angl. mixed-integer programming), ktorý vieme riešiť konvenčnými numerickými nástrojmi, ako je napríklad GUROBI[1]. Aj keď na prvý pohľad takáto verzia premeny dvojúrovňového optimalizačného problému vyzerá nádejne, treba si uvedomiť, že množstvo regiónov ktoré vzniká pri riadení procesov typu chemického reaktora sú prinajmenšom tisíce. Takáto štruktúra optimalizačného problému z neho robí prakticky neriešiteľný problém.

Druhý prístup k reformulácii dvojúrovňových problémov má vo výsledku tiež charakter zmiešaného celočíselného programovania, avšak s podstatne menším počtom binárnych premenných. Vnútorný optimalizačný problém je totiž možné priamo pomocou Karush-Kuhn-Tuckerových podmienok prepísať, čím získame sústavu rovníc, ktoré charakterizujú optimum daného vnútorného MPC problému. Ako prvé prepíšeme kvadratický optimalizačný problém do štandardnej formy kvadratického programovania:

$$\min_{U_i} \ \frac{1}{2} U_i^\intercal H_i U_i + \theta_i^\intercal F_i U_i + a_i^\intercal U_i \tag{2.17a}$$

$$\text{v.n.} \ \ G_i U_i \leq V_i + E_i \theta_i, \tag{2.17b}$$

[1] www.gurobi.com

pre ktorý potom môžeme rozpísať sériu KKT podmienok

$$HU^\star + F^\intercal\theta + a \ + G^\intercal\lambda^\star = 0, \tag{2.18a}$$
$$GU^\star - V \ - E\theta \leq 0, \tag{2.18b}$$
$$\lambda^\star \geq 0, \tag{2.18c}$$
$$\lambda_s^\star(G^{[s]}U^\star - V^{[s]} - E^{[s]}\theta) = 0, \quad \forall s \in \mathbb{N}_1^{n_\mathrm{G}}. \tag{2.18d}$$

Pre jednoduchosť matematického zápisu sme prestali uvažovať spodný index i, ktorý reprezentuje o ktorý vnútorný MPC regulátor ide. Rovnice v bloku (2.18) sú v poradí: podmienky stacionarity, primárna zlučiteľnosť, duálna zlučiteľnosť a podmienky doplnkovej voľnosti. Premenná $\lambda^\star$ je vektorom optimálnych hodnôt Lagrangeových násobičov, pričom index n_G znázorňuje celkový počet ohraničení v tvare nerovnosti. Ak by sme rovnicu (2.18d) ponechali v nezmenenom tvare a dosadili ju za model regulátora v MPC-RG formulácii, v konečnom dôsledku by sme dostali istú formu kvadratického ohraničenia, pretože nielen vektor U_i je optimalizovanou premennou, ale rovnako aj λ_i. Preto sme pristúpili k zápisu ohraničenie doplnkovej voľnosti pomocou nasledujúceho vzťahu

$$[G^{[s]}U^\star - V^{[s]} - E^{[s]}\theta < 0] \implies [\lambda_s^\star = 0], \tag{2.19}$$

ktorý sa dá pomocou *big-M* metódy a pravidiel logických operácií [Wil93] preformulovať na

1. $[f(z) < 0] \implies [\delta = 1]$ ak a iba ak $f(z) \geq Z_{\min}\delta$,
2. $[\delta = 1] \implies [f(z) = 0]$ ak a iba ak $Z_{\min}(1-\delta) \leq f(z) \leq Z_{\max}(1-\delta)$,
3. $[\delta = 1] \implies [f(z) \leq 0]$ ak a iba ak $f(z) \leq Z_{\max}(1-\delta)$,

Takto vieme získať zmiešaný celočíselný optimalizačný pod-problém v tvare

$$HU^\star + F^\intercal\theta + a \ + G^\intercal\lambda^\star = 0, \tag{2.20a}$$
$$GU^\star - V \ - E\theta \leq 0, \tag{2.20b}$$
$$\lambda^\star \geq 0, \tag{2.20c}$$
$$GU^\star - V \ - E\theta \geq Z_{\min}\delta, \tag{2.20d}$$
$$\lambda^\star \geq Z_{\min}(\mathbf{1}_{n_\mathrm{G}} - \delta), \tag{2.20e}$$
$$\lambda^\star \leq Z_{\max}(\mathbf{1}_{n_\mathrm{G}} - \delta), \tag{2.20f}$$

ktorý reprezentuje KKT podmienky vnútorného MPC problému. Následne môžeme túto

sústavu rovníc dosadiť do MPC-RG stratégie (2.3) čím dostaneme

$$\min_{\lambda_{i,j},\delta_{i,j},u_{i,j},w_{i,j}} \sum_{j=0}^{\overline{N}-1} \sum_{i=1}^{M} \ell_{\mathrm{M},i}\left(r_{i,j}, y_{i,j}, w_{i,j}\right), \tag{2.21a}$$

$$\text{v.n.} \quad H_i U^\star_{i,j} + F^\intercal \theta_{i,j} + a_i + G_i^\intercal \lambda_{i,j} = 0, \tag{2.21b}$$

$$G_i U^\star_{i,j} - V_i - E_i \theta_{i,j} \leq 0, \tag{2.21c}$$

$$\lambda_{i,j} \geq 0, \tag{2.21d}$$

$$G_i U^\star_{i,j} - V_i - E_i \theta_{i,j} \geq Z_{\min} \delta_{i,j}, \tag{2.21e}$$

$$\lambda_{i,j} \leq Z_{\max}(\mathbf{1}_{n_\mathrm{G}} - \delta_{i,j}), \tag{2.21f}$$

$$u^\star_{i,j} = \Phi U^\star_{i,j}, \tag{2.21g}$$

$$x_{i,j+1} = A_i x_{i,j} + B_i u^\star_{i,j}, \tag{2.21h}$$

$$y_{i,j} = C_i x_{i,j} + D_i u^\star_{i,j}, \tag{2.21i}$$

$$\bar{x}_j \in \overline{\mathcal{X}}, \tag{2.21j}$$

$$\bar{u}^\star_j \in \overline{\mathcal{U}}, \tag{2.21k}$$

$$\bar{y}_j \in \overline{\mathcal{Y}}, \tag{2.21l}$$

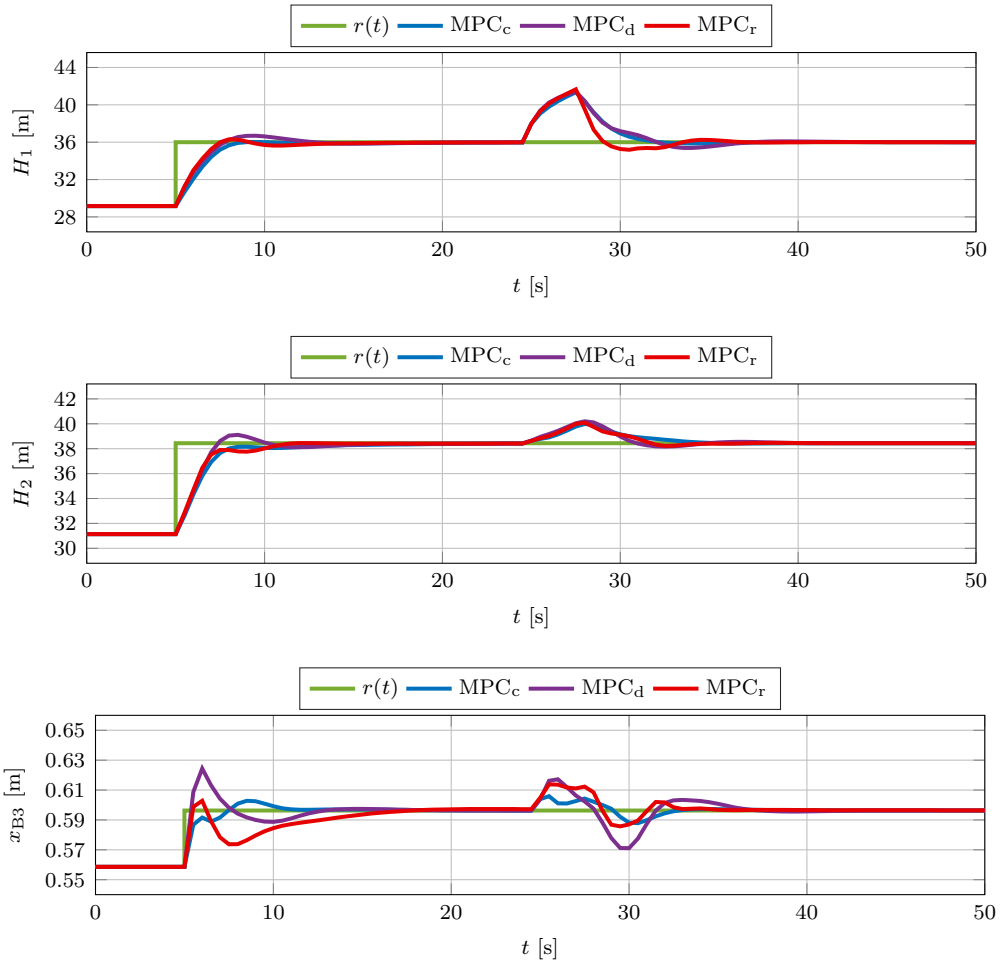
$$\delta_{i,j} \in \{0,1\}^{n_{\mathrm{G},i}}. \tag{2.21m}$$

Aj keď štruktúra výsledného optimalizačného problému má charakter zmiešaného celočíselného programovania, tak tento typ problémov vieme riešiť numericky v rozumnom čase pomocou nástrojov ako GUROBI alebo CPLEX. Oproti prvému zmienenému prístupu na riešenie dvojúrovňového optimalizačného problému pomocou explicitného riešenia ušetríme v tomto druhom prístupe veľmi veľké množstvo binárnych premenných, pretože ich potrebujeme len toľko, koľko máme ohraničení v tvare nerovnosti. A tých je spravidla rádovo menej, ako počet regiónov v parametrickom riešení.

Tabuľka 2.1: Numerické vyjadrenie dosiahnutia kvalitatívneho zlepšenia.

Riadiaca schéma	J_{ISE}	$\Delta\ J_{\mathrm{ISE}}$ [%]
Centralizovaný prístup	$7.2 \cdot 10^4$	0.0
Decentralizovaný prístup	$8.1 \cdot 10^4$	11.8
Zapojenie s MPC koordinátorom	$7.3 \cdot 10^4$	1.2

Výhody a kvalitatívne porovnanie využitia nadradenej MPC stratégie pre koordináciu viacerých nízkoúrovňových MPC regulátorov demonštrujeme na prípadovej štúdii z [KK19b], ktorá porovnáva riadenie dvoch sériovo zapojených reaktorov s odparovačom, tak ako je zobrazené na schéme na Obr. 2.6. Keďže zdrojová knižná autorská práca nie je distribuovaná v rámci tejto habilitačnej práce z rozsahových dôvodov, reprodukujeme konkrétne výsledky v tejto práci. Zobrazujeme tri série časových priebehov:

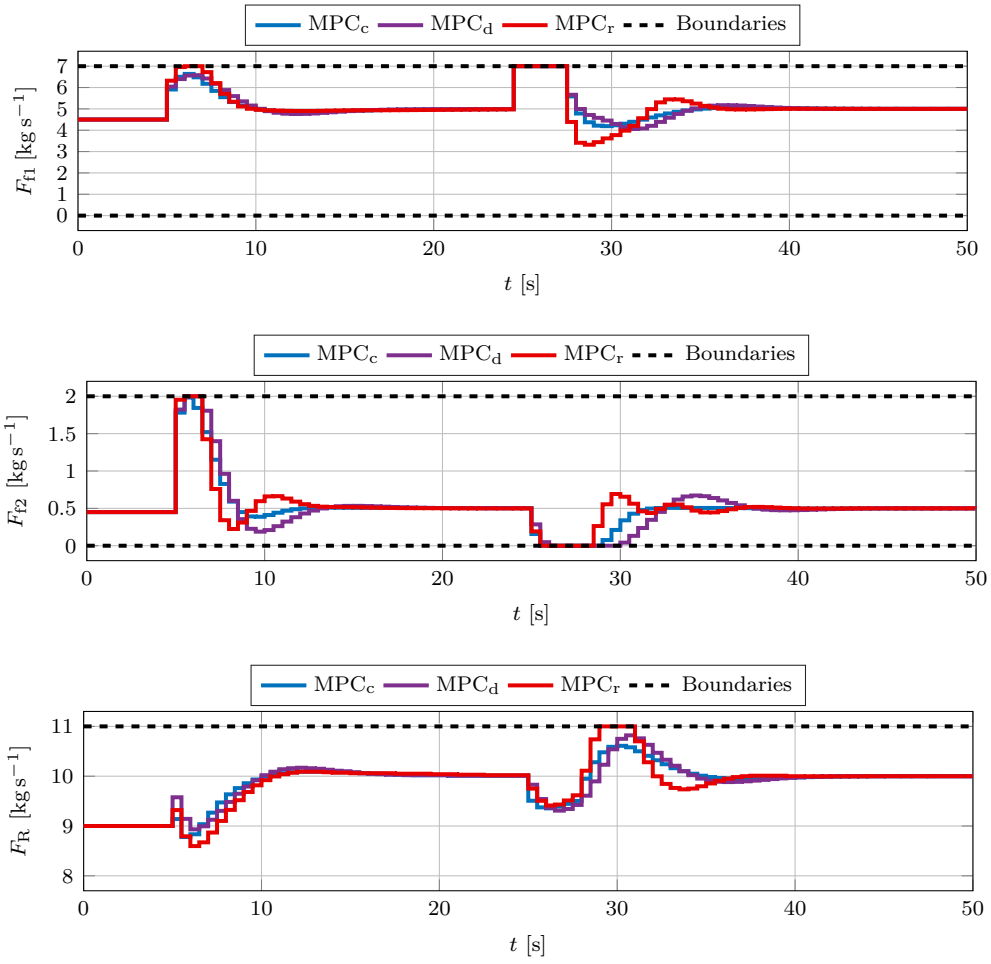


Obr. 2.7: Časová závislosť riadených veličín.

- na Obr. 2.7: riadenie výšky hladín reakčných zmesí v reaktoroch $H_{1,2}$ a koncentráciu produktu $x_{\mathrm{B},3}$ extrahovaného z odparovača,
- na Obr. 2.8: priebehy akčných veličín, ktorými sú prietoky v rámci jednotlivých častí procesu,
- na Obr. 2.9: optimálnu žiadanú hodnotu pre nízkoúrovňové MPC regulátory nastavené pomocou MPC koordinátora.

Prípadová štúdia porovnáva tri scenáre riadiacej stratégie:

1. stratégia označená v grafoch ako $\mathrm{MPC_d}$ predstavuje priame riadenia jednotlivých častí procesu separátnymi nekoordinovanými MPC regulátormi,



Obr. 2.8: Časová závislosť riadiacich veličín.

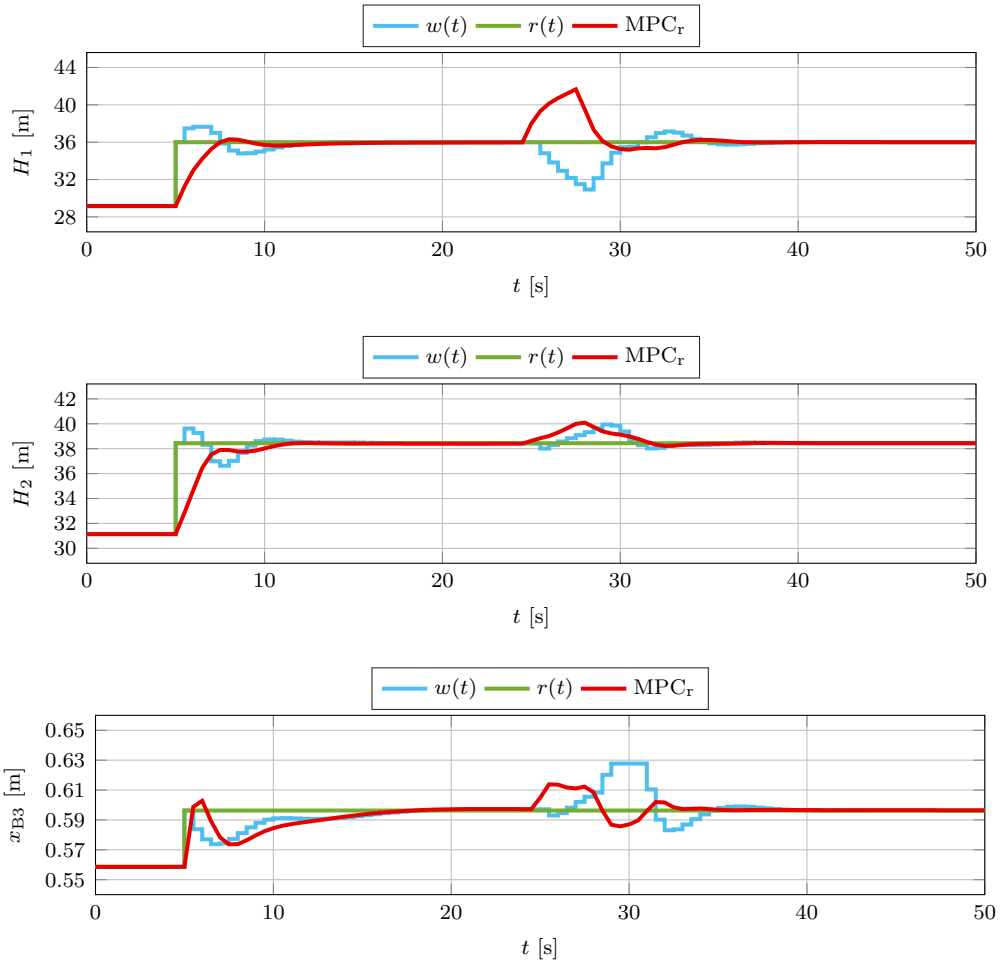
2. označenie MPC$_c$ predpokladá, že by sme celý systém riadili pomocou jedného MPC regulátora,

3. a výsledky označené MPC$_r$ prezentujú ako sa bude správať celý systém pod dohľadom MPC koordinátora.

Je potrebné poznamenať, že MPC$_c$ je považovaná za ideálnu a v praxi nerealistickú, pretože málokedy sa naozaj stane, že iba jeden regulátor je zaradený do riadenia väčšieho celku procesov. Z numerického pohľadu sme kvalitu riadenia vyhodnotili pomocou

$$J_{\mathrm{ISE}} = \sum_{k=1}^{T_{\mathrm{f}}} (r_k - y_k)^{\mathsf{T}} Q (r_k - y_k), \tag{2.22}$$

pričom váhová matica Q je zvolená tak, aby príspevok každého člena v rámci vektora



Obr. 2.9: Modulovaná referencia v prípade použitia MPC koordinátora.

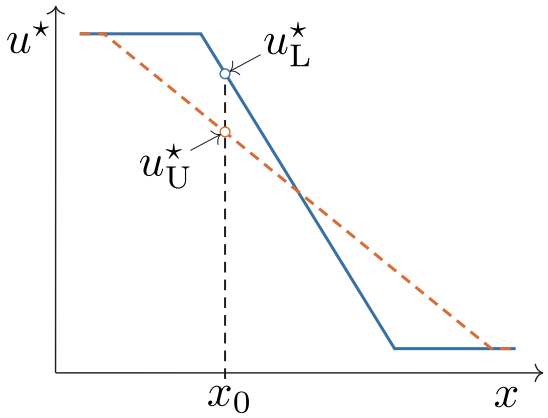
$(r_k - y_k)$ bol rádovo rovnaký. Konkrétne numerické výsledky prezentujeme v tabuľke 2.1, kde môžeme vidieť, že z kvalitatívneho hľadiska použitie MPC koordinátora dosahuje skoro rovnaké výsledky ako idealizovaný priamy scenár $\mathrm{MPC_d}$.

2.3 Syntéza za-behu laditeľného MPC

Základnou motiváciou riešiť problém priebežne laditeľného regulátora vznikla práve z riešenia problematiky nadradených MPC regulátorov. Vieme, že pokiaľ uvažujeme návrh MPC regulátora, tak požadujeme aby bola fixná jeho štruktúra, teda model riadeného systému ako aj dĺžka predikčného horizontu a hodnota váhových matíc. V prípade, že sa zmení čo i len jedna z vyššie vymenovaných súčasti, je potrebné celý optimalizačných problém zostaviť odznova. Najmä v priemyselných aplikácia sa stretávame so situáciou,

keď priamo operátorom je dovolené meniť parametre napríklad PID regulátorov v určitých rozsahoch. Na strane MPC riadenia sa v tomto kontext jedná o priebežné zmeny hodnôt váhových matíc v účelovej funkcii. Práve tento aspekt návrhu MPC regulátorov je načrtnutý v tejto sekcii, pričom konkrétne technické riešenie spolu s matematickým dôkazom prikladáme v tejto habilitačnej práci a sú publikované v časopiseckej práci autora [OK22], ktorá čerpá aj z [KK18].

Zmena parametrov na strane MPC riadenia je problematická z niekoľkých dôvodov. Ako prvé musíme pripomenúť možnú stratu rekurzívnej riešiteľnosti, alebo schopnosť stabilizovať systém, ak sa bez hlbšieho zamyslenia počas riadenia zmení váhová matica na stavy. Ďalším problematickým momentom je zmena predikčného modelu, ktorá priamo vplýva na štruktúru optimalizačného problému spolu s dĺžkou predikčného horizontu. Aj keď pri online riešenie tejto úlohy si môžeme dovoliť robiť niektoré zmeny v MPC optimalizačnom probléme bez nejakých veľkých negatív, naopak v prípade explicitného riešenia to nie je vôbec možné. Akákoľvek zmena štruktúry optimalizačného problému vedie na okamžitú nutnosť znova skonštruovať zákon riadenia v explicitnom tvare. Výsledky prezentované najme v práci [OK22] ukazujú, že sme schopní navrhnúť také explicitné riešenie, v ktorom môžeme počas riadenia meniť hodnotu jeden z váhovej matíc bez nutnosti jeho rekonštrukcie pri jej zmene.



Obr. 2.10: Určenie akčného zásahu medzi dvoma hraničnými explicitnými regulátormi.

V tejto práci načrtneme možnosť zostrojenia dvoch hraničných explicitných regulátorov pre hraničné hodnoty váhových matíc. Keďže získame dva zákony riadenia v explicitnom tvare, tak pre konkrétnu hodnotu meraného parametra (začiatočnej podmienky), vieme získať dve hraničné hodnoty akčných zásahov. Následne medzi týmito dvoma hraničnými hodnotami akčného zásahu budeme interpolovať podľa zmeny váhovej matice. Ak by sa jednalo o jednorozmerný systém, výpočet takéhoto interpolovaného akčného zásahu je znázornený na Obr. 2.10.

Pre potreby odvodenia za-behu laditeľného explicitného MPC s garanciami stability

a rekurzívnej riešiteľnosti, uvedieme základnú formuláciu hraničných regulátorov, a to

$$\min_{u_0,u_1,\ldots,u_{N-1}} \quad x_N^\intercal P_\mathrm{L} x_N + \sum_{k=0}^{N-1} \left(x_k^\intercal Q_\mathrm{L} x_k + u_k^\intercal R_\mathrm{L} u_k\right) \tag{2.23a}$$

$$\text{s.t.:} \quad x_{k+1} = Ax_k + Bu_k, \tag{2.23b}$$

$$u_k \in \mathcal{U}, \tag{2.23c}$$

$$x_k \in \mathcal{X}, \tag{2.23d}$$

$$x_N \in \mathcal{T}_\mathrm{L}, \tag{2.23e}$$

$$x_0 = \theta(t), \tag{2.23f}$$

$$k = 0, 1, \ldots, N-1, \tag{2.23g}$$

a

$$\min_{u_0,u_1,\ldots,u_{N-1}} \quad x_N^\intercal P_\mathrm{H} x_N + \sum_{k=0}^{N-1} \left(x_k^\intercal Q_\mathrm{H} x_k + u_k^\intercal R_\mathrm{H} u_k\right) \tag{2.24a}$$

$$\text{s.t.:} \quad (2.23b), (2.23c), (2.23d), (2.23f), (2.23g), \tag{2.24b}$$

$$x_N \in \mathcal{T}_\mathrm{H}. \tag{2.24c}$$

Operátor resp. užívateľ má za cieľ meniť hodnotu váhových matíc, čiže v skutočnosti by chcel vyriešiť problém v tvare

$$\min_{u_0,u_1,\ldots,u_{N-1}} \quad x_N^\intercal \widetilde{P} x_N + \sum_{k=0}^{N-1} \left(x_k^\intercal \widetilde{Q} x_k + u_k^\intercal \widetilde{R} u_k\right) \tag{2.25a}$$

$$\text{s.t.:} \quad (2.23b), (2.23c), (2.23d), (2.23f), (2.23g), \tag{2.25b}$$

$$x_N \in \widetilde{\mathcal{T}}, \tag{2.25c}$$

kde platí, že hodnoty vybraných váhových matíc musia spĺňať nasledovné podmienky

$$\widetilde{R} \;=\; (\rho - 1)R_\mathrm{L} + \rho\, R_\mathrm{H}, \quad 0 \le \rho \le 1, \tag{2.26a}$$

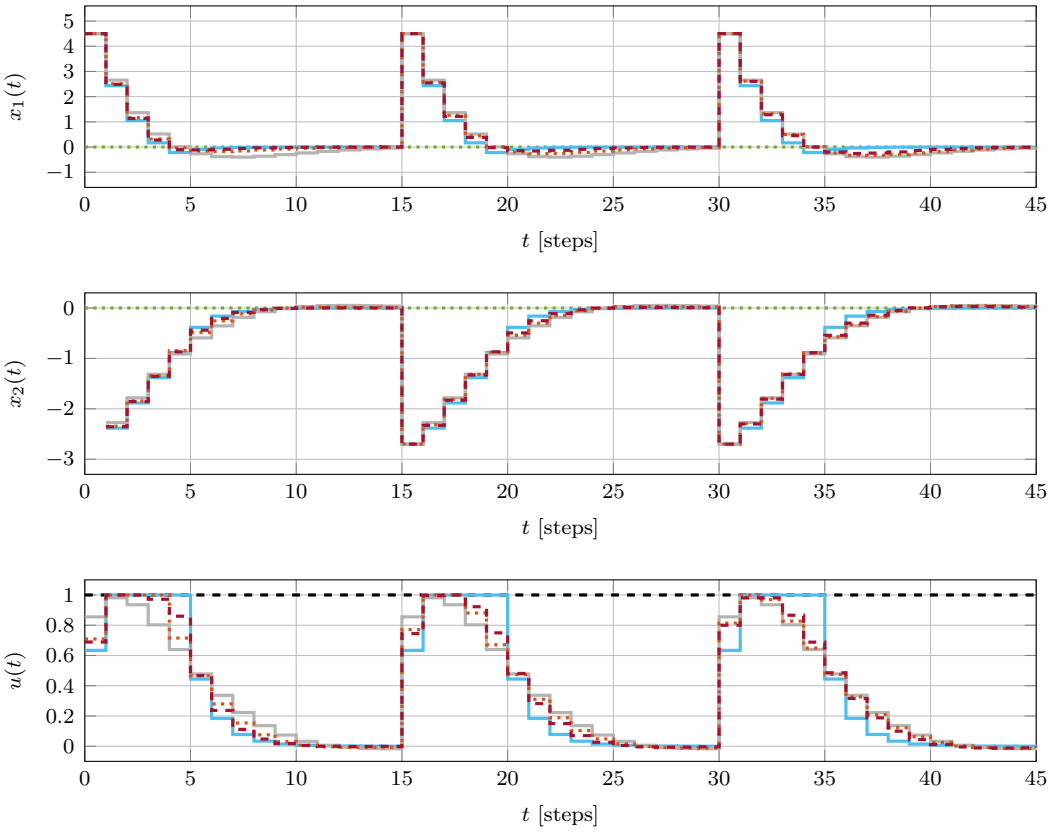
$$\widetilde{Q} \;=\; (\phi - 1)Q_\mathrm{L} + \phi\, Q_\mathrm{H}, \quad 0 \le \phi \le 1. \tag{2.26b}$$

Ak budeme predpokladať, že sa bude meniť iba jedna z váhových matíc, tak potom hodnota interpolovaného akčného zásahu je určená ako

$$\widetilde{u} \;=\; (\rho - 1)u_\mathrm{L} + \rho\, u_\mathrm{H}, \qquad 0 \le \rho \le 1, \tag{2.27}$$

kde hodnoty u_L, u_H sú optimálne akčné zásahy pre konkrétnu počiatočnú podmienku získané pomocou hraničných regulátorov. Garancie stability a rekurzívnej riešiteľnosti, zabezpečíme pomocou vhodného nastavenia terminálnej penalizácie a voľby ohraničení na koncový stav. Dôkaz tohto tvrdenia je uvedený v priloženej publikácií [OK22]. Simulačný scenár, ktorým dokazujeme vlastnosti za-behu laditeľného MPC regulátora, je založený na modeli dvojitého-integrátora, ktorý excitujeme zmenou začiatočnej podmienky. Počas

troch skokových zmien, ktoré sú zobrazené na Obr. 2.11, meníme postupne hodnotu parametra $\rho = 0.25, 0.50, 0.75$, pričom rozsah váhovej matice penalizujúcej akčné zásahy je $R_{\mathrm{L}} = 0.5$, $R_{\mathrm{H}} = 10.0$. Z uvedenej simulácie ako aj z výsledkov v priloženej publikácií [OK22] vidno, že riešenie pomocou interpolovaného zákona riadenia z rovnice (2.27) je veľmi blízke skutočnému optimálnemu riešeniu. Zároveň sme týmto prístupom odbúrali nutnosť konštruovať explicitný regulátor v prípade potreby zmeny váhových matíc.



Obr. 2.11: Riadenie za-behu laditeľným explicitným regulátorom $\widetilde{u}$ je znázornené červenou čiarkovanou krivkou. Modrá a šedá farba znázorňujú riadenie pomocou hraničných regulátorov u_{L}, resp. u_{H}, pričom optimálna hodnota akčného zásahu vzhľadom na aktuálnu hodnotu váhových matíc u_{opt} je znázornená oranžovou.

Kapitola 3

Metódy strojového učenia v prediktívnom riadení

3.1 Inicializácia metódy aktívnych množín pomocou strojového učenia

Drvivá väčšina optimalizačných problémov v riadení procesov, najme teda tých, ktoré majú charakter prediktívneho riadenia v tvare kvadratického optimalizačného problému, sa rieši pomocou metódy aktívnych množín (z angl. active set method - ASM). Jej výhodou je, že aj pri komplexnejších problémoch vyžaduje relatívne malý počet iterácií na nájdenie optimálneho riešenia [Fle13]. Uvažujme, že uvažovaný problém optimálneho riadenia v tvare MPC je daný ako

$$\min_{U} \; 1/2\, U^\intercal H U + \theta^\intercal F U \tag{3.1a}$$

$$\text{s.t. } GU \leq E\theta + w, \tag{3.1b}$$

pričom, ide o vyjadrenie MPC problému podobne ako sme definovali v predošlej kapitole, konkrétne (2.17).

Základným princípom tejto metódy je hľadanie veľkosti smeru (vektora) Δ podľa ktorého klesá hodnota účelovej funkcie. V každej iterácií tejto metódy sa rieši kvadratický problém s ohraničeniami v tvare rovnosti, ktorý pre problém optimálneho riadenia vyzerá nasledovne

$$\min_{\Delta} \; \frac{1}{2} (U + \Delta)^\intercal H (U + \Delta) + \theta^\intercal F (U + \Delta) \tag{3.2a}$$

$$\text{s.t. } G_{\mathcal{A}} (U + \Delta) = E_{\mathcal{A}} \theta + w_{\mathcal{A}}, \tag{3.2b}$$

pričom, ohraničenia v tvare rovnosti sú vybrané na základe aktuálnej množiny aktívnych

ohraničení $\mathcal{A}$. Tento problém sa dá jednoducho riešiť pomocou KKT podmienok

$$\begin{bmatrix} \Delta \\ \lambda \end{bmatrix} = \begin{bmatrix} H & G_{\mathcal{A}}^{\intercal} \\ G_{\mathcal{A}} & 0 \end{bmatrix}^{-1} \begin{bmatrix} -HU - F^{\intercal}\theta \\ 0 \end{bmatrix}, \tag{3.3}$$

pričom ak niektorý z Langrangeových násobičov λ nadobudnú záporné hodnoty, tak je nutné upraviť množinu $\mathcal{A}$. Pre úplnosť uvádzame plnohodnotný algoritmus metódy aktívnych množín

Algorithm 1 Metóda aktívnych množín

Urči počiatočnú $\mathcal{A}_0$ a zlučiteľné riešenie U_0
Nastav $\mathcal{A} \leftarrow \mathcal{A}_0$, $U \leftarrow U_0$
while nie je dosiahnutá konvergencia **do**
  Vyrieš (3.3) a získaj Δ, λ
  if $\|\Delta\| = 0$ **then**
    if $\lambda \geq 0$ **then**
      return $U^\star = U$
    else
      $\mathcal{A} \leftarrow \mathcal{A} \setminus \{i^\star\}$ vzhľadom na $i^\star = \arg\min_{i \in \mathcal{A}} \lambda_i$
    end if
  else
    Urči β_j vzhľadom na $\beta_j = \dfrac{E_j\theta + w_j - G_jU}{G_j\Delta}$ pre $j \notin \mathcal{A}$, $G_j\Delta > 0$
    Nastav $\alpha = \min\{1, \beta_j\}$
    Nastav $U \leftarrow U + \alpha\Delta$
    if $\alpha < 1$ **then**
      $\mathcal{A} \leftarrow \mathcal{A} \cup \{j^\star\}$ kde $j^\star = \arg\min_j \beta_j$
    end if
  end if
end while

Základným faktorom pri počte iterácií, ktoré je potrebné vykonať na nájdenie optima, je samozrejme inicializácia optimalizačnej metódy. Drvivá väčšina aplikácií tejto metódy uvažuje tzv. studený štart, čiže ASM metódu inicializuje prázdnou množinou aktívnych ohraničení a zároveň počiatočný zlučiteľný bod uvažuje ako nulový vektor.

Keďže pri aplikácií prediktívnych regulátorov riešime optimalizáciu periodicky, každú periódu vzorkovania, je vhodné prehodnotiť inicializáciu iteračnej metódy. Ak by sme totiž vhodne zvolili $\mathcal{A}_0$, dokonca ak by ju zvolili optimálne vzhľadom na hodnotu počiatočnú podmienky, tak sa výpočet optimálneho akčného zásahu $U^\star$ zredukuje na jednu iteráciu. Určiť ale dopredu zoznam aktívnych ohraničení by znamenalo mať k dispozícií explicitné riešenie daného problému, čo je však nepraktické pre problémy väčších rozmerov.

V priloženej časopiseckej publikácií [KKK19] je ukázané, ako získať dobrý odhad $\mathcal{A}_0$ vzhľadom na hodnotu počiatočnej podmienky. Tento odhad získame, tak, že využijeme metódy strojového učenia, ktoré nám natrénujú tzv. prediktor, teda funkciu, ktorej vstupom bude hodnota počiatočnej podmienky θ resp. x_0 v (3.1) a výstupom bude odhad inicializácie $\mathcal{A}_0$. Keďže v konečnom dôsledku nenahrádzame ASM pri riešení prediktívneho riadenia, máme všetky kvalitatívne garancie, ktoré prediktívne riadenia poskytuje. Hlavným benefitom, tak ako uvádzame v publikácií, je že sme týmto spôsobom schopní zredukovať počet iterácií nutných na výpočet optimálneho akčného zásahu v desiatkach percent, v niektorých prípadoch až 20-násobne.

Celkový mechanizmus na získanie a použitie takýchto prediktorov inicializačných parametrov ASM metódy je rozdelené na dve časti. Prvá z nich, prípravná resp. "offline" fáza pozostáva:

1. Vygenerovať množinu bodov počiatočných podmienok θ.
2. Určiť zoznamy aktívnych ohraničení pre jednotlivé body počiatočných podmienok.
3. Natrénovať mapovaciu funkciu κ, ktorej vstupom je počiatočná podmienka a výstupom zoznam aktívnych ohraničení.

Implementačná fáza resp. "online" fáza sumarizuje využitie v uzavretom regulačnom obvode:

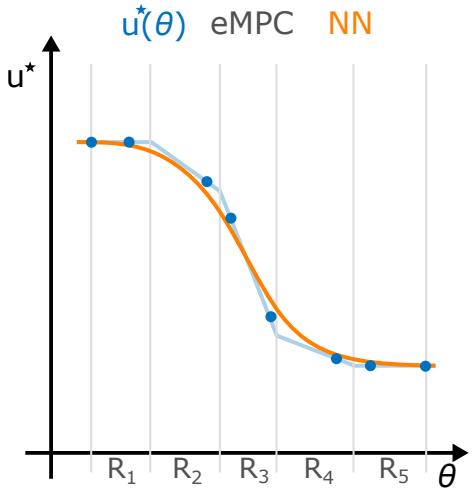
1. Získanie stavového merania z riadeného procesu.
2. Vyhodnotenie prediktora κ.
3. Inicializácia ASM metódy výsledkom z κ.
4. Určenie optimálneho akčného zásahu a jeho implementácia v riadenom systéme.

Takýto spôsob urýchlenia výpočtu optimálneho akčného zásahu nám tým pádom umožňujú implementáciu numerických metód aj na mikroprocesorových platformách, ktoré nedisponujú veľkým výkonom. Keďže výrazne znížime počet iterácií, suplujeme znížený výpočtový výkon, čiže stále môžeme optimálnym spôsobom riadiť systémy vyžadujúce nízku periódu vzorkovania.

3.2 Aproximácia MPC pomocou neurónových sietí

Medzi populárne vedecké smery momentálne patrí skoro akýkoľvek prienik metód strojového učenia do iných sfér vedy, riadenie procesov, resp. syntézu regulátorov nevynímajúc. V tomto ohľade predstavíme niekoľko prvotných výsledkov, ktoré ukazujú, že strojové učenie, konkrétne neurónové siete sú schopné kompletne nahradiť prediktívny regulátor.

Základné predpoklady tohto vedeckého smeru vychádzajú z nemožnosti skonštruovať explicitné riadenie pre systémy s veľkým parametrickým priestorom, resp. dlhým predikčným horizontom. Predkladaná habilitačná práca si dáva za cieľ ukázať, že sme schopní zostaviť explicitnú formu MPC aj pre systémy s veľkým počtom stavov a optimalizovaných premenných. Nebudeme však riešiť problém parametrickej optimalizácie, ale zákon riadenia "naučíme".



Obr. 3.1: Aproximácia explicitného riešenia pomocou neurónovej siete.

Práve neurónové siete, sú schopné v princípe aproximovať akúkoľvek funkciu, čiže aj nekonvexný zákon riadenia [LK18]. Výhodou využitia neurónových sietí je najmä to, že nám odpadá nutnosť uchovať si informácie o regiónoch, čím drasticky znížime pamäťovú náročnosť explicitného riešenia (viď Obr. 3.1). Úlohou je teda navrhnúť také metódy pri získavaní parametrov neurónovej siete, aby sme získali čo najvervnejšiu reprezentáciu explicitného riešenia.

Neurónová sieť pozostáva zo vstupnej vrstvy, skrytých vrstiev a z výstupnej vrstvy, ako je znázornené na Obr. 3.2. Každá z vrstiev obsahuje niekoľko neurónov, pričom každý z nich je reprezentovaný aktivačnou funkciou, napríklad sigmoidou

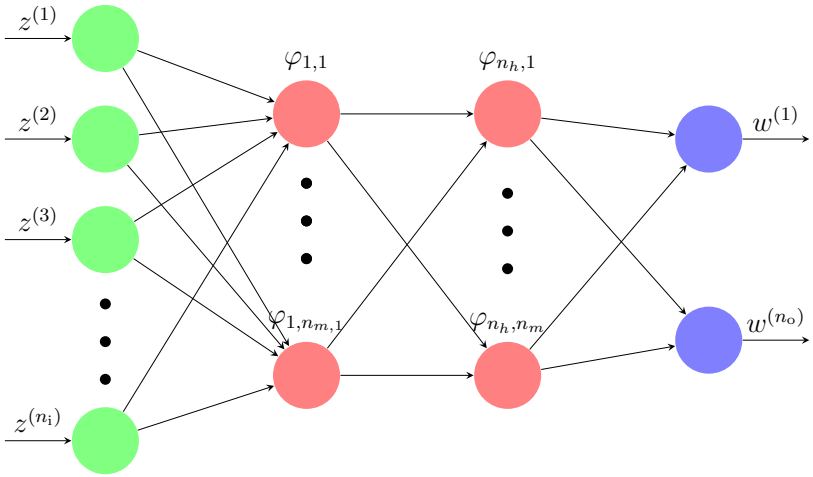
$$\varphi(\alpha, z) = \frac{2}{1 + \mathrm{e}^{\alpha z}} - 1, \tag{3.4}$$

alebo po častiach lineárnou funkciou (z angl. rectified linear unit - ReLU)

$$\mathrm{ReLU}(\alpha, z) = \max(0, \alpha \cdot z \ + b), \tag{3.5}$$

kde z značí vstupný vektor a parameter α je vektor váh, ktoré je potrebné natrénovať.

Postup získania aproximácie má podobne ako v prípade predošlej sekcie, dve fázy. V prvej fáze je potrebné pripraviť vstupy na trénovanie neurónovej siete a natrénovať



Obr. 3.2: Ilustrácia neurónovej siete, kde je zelenou vyznačená vstupná vrstva neurónov, červenou sú označená skryté vrstvy a modrá znázorňuje výstupnú vrstvu.

neurónovú sieť. Druhá fáza predpokladá implementáciu v spätnoväzbovom riadení, kde výhody explicitne daného zákona riadenia odbúrajú akékoľvek potrebu riešiť optimalizáciu. Prvá fáza pozostáva

1. Zostrojenie prediktívneho regulátora.
2. Vygenerovať množinu bodov počiatočných podmienok θ a k nim prislúchajúcich akčných zásahov $u_0^\star$.
3. Zvoliť parametre neurónovej siete, t.j. počet vrstiev, počet neurónov na vrstvách a typy aktivačných funkcií.
4. Natrénovať váhové parametre jednotlivých aktivačných funkcií, čím získame zákon riadenia $f_{\text{NN}}(\theta)$.

Implementačná fáza resp. "online" fáza opäť sumarizuje využitie v uzavretom regulačnom obvode:

1. Získanie stavového merania z riadeného procesu θ.
2. Vyhodnotenie explicitného zákona riadenia $u_{\text{NN}} = f_{\text{NN}}(\theta)$.
3. Implementácia aproximovaného akčného zásahu.

Vlastnosti takého riadenia budeme demonštrovať na úlohe sledovania žiadanej hodnoty pre systém dvojitého integrátora. V prvom kroku zostrojíme regulátor podľa predpisu v rovnici (2.1). Následne si pripravíme náhodne distribuovaných 5000 bodov v stavovom

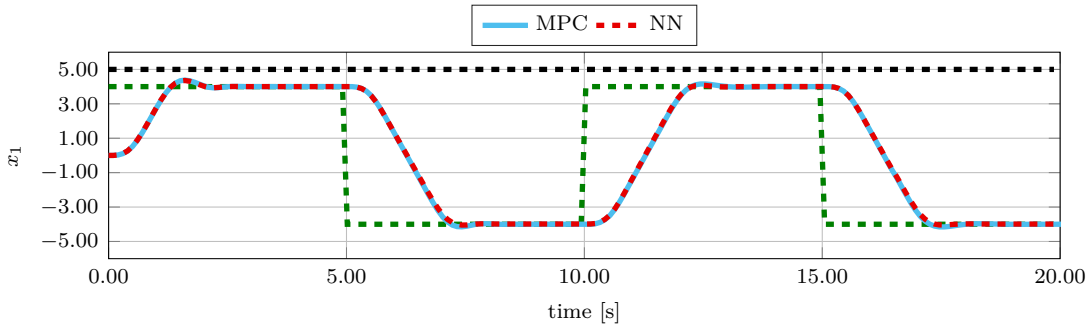
priestore modelu dvojitého integrátora. Pre každý z bodov v rámci týchto počiatočných podmienok získame hodnotu akčného zásahu. Takto zostrojená množina stavov a optimálnych akčných zásahov vstúpi do trénovacej fázy. Neurónová sieť pozostávala z troch skrytých vrstiev, pričom každá z nich mala 10 neurónov a aktivačné funkcie boli zvolené v tvare ReLU.

Kvalitu riadenia pomocou aproximovaného zákona riadenia zobrazujeme na Obr. 3.3. Z porovnania je zrejmé, že regulátor v tvare neurónovej siete je schopný s veľmi veľkou presnosťou riadiť systém tak, ako pôvodné MPC. Najväčšou nevýhodou je pochopiteľne garancia splnenia ohraničení a garancia stability. V tabuľke 3.1 zobrazujeme tri numericky vyhodnotené kvalitatívne parametre. Integrálne kritérium ISE z rovnice (2.22), počet porušení ohraničení $\mathcal{N}(\cdot)$ a maximálna veľkosť porušenia ohraničenia v absolútnej hodnote $\mathcal{V}_{\mathrm{m}}(\cdot)$.

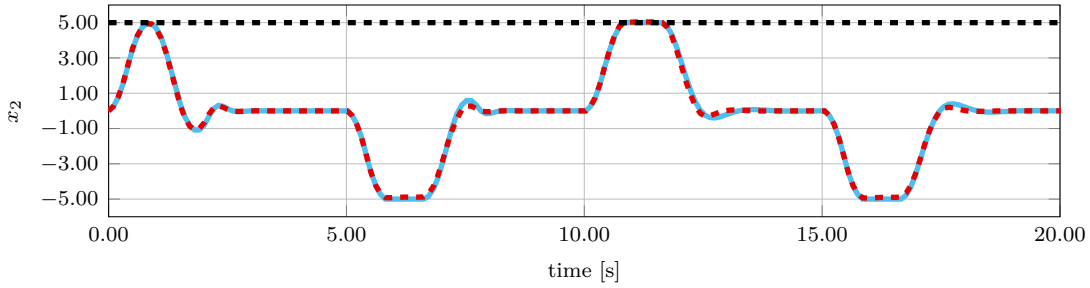
Tabuľka 3.1: Qualitative evaluation of control performance.

	J_{ISE}	porušenie ohraničení			
		$\mathcal{N}(x_2)$	$\mathcal{V}_{\mathrm{m}}(x_2)$	$\mathcal{N}(\Delta u)$	$\mathcal{V}_{\mathrm{m}}(\Delta u)$
MPC	574	0	0	0	0
NN	580	19	0.140	29	0.036

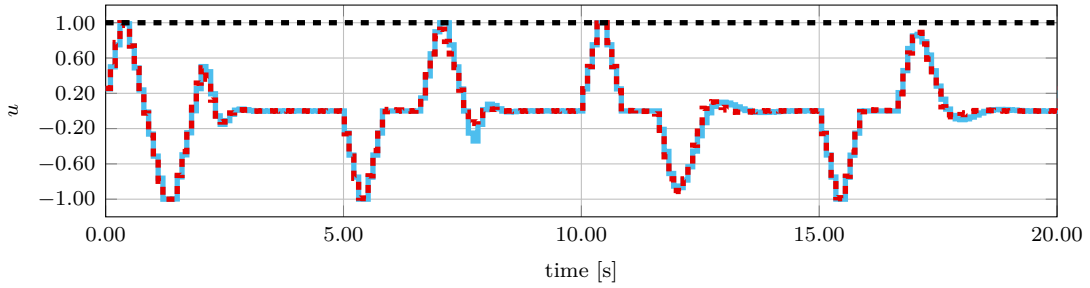
Keďže ide o nový výskumný smer, tak tieto otázky ešte neboli dostatočne vyriešené. Pri uvažovaní všetkých aspektov implementácie akéhokoľvek riadenia na laboratórny proces alebo aj priamo v technologickej prevádzke, teoretické garancie sú mnohokrát druhoradé, a extenzívna simulačná prípadová štúdia postačuje ako dôkaz [KK19a, LKFM20, LKKM19].



(a) Priebeh stavu x_1.



(b) Priebeh stavu x_2.



(c) Piebeh akčných zásahov.

Obr. 3.3: Riadenie pomocou aproximovaného zákona riadenia pomocou neurónových sietí. Modrá krivka reprezentuje optimálne priebehy, pričom červená značí aproximované riadenie.

Kapitola 4

Záver a diskusia

Predložená práca sa sústreďuje na nové aspekty využitia prediktívneho riadenia. Nosným pilierom tejto habilitačnej práce je dlhodobá práca autora v oblasti návrhu nadradených regulátorov založených na báze MPC. V tejto partikulárnej oblasti sme ukázali ako navrhnúť nadradené regulátory pre rôzne typy vnútorných riadiacich slučiek. Okrem očividnej výhody tohto prístupu k riadeniu, ktorá umožňuje zvýšiť kvalitatívne parametre riadenia už existujúcich riadiacich systémov, tak ide o systematizovaný prístup k syntéze tohto typu regulátorov, ktorý zahŕňa všetky tri kľúčové vlastnosti dobrého regulátora:

1. penalizácia optimalizačného kritéria,
2. predikcia správania sa na základe modelu systému,
3. integrácia technologických ohraničení priamo do návrhu nadradených regulátorov.

Okrem vyššie spomenutého využitia MPC regulátorov, sa predložená práca venuje aj návrhu explicitných MPC regulátorov, ktorým je možné "za jazdy" zmeniť hodnotu váhových matíc. Riešenie tohto problému výrazne posunulo vpred možnosti nasadenia explicitných regulátorov do praxe, pretože dramaticky znižuje nutnosť rekonštrukcie explicitného riešenia v prípade zmeny váhových matíc.

Posledná časť tejto habilitačnej práce je venovaná novému výskumnému smeru autora, ktorý má za cieľ hlbšie prepojiť možnosti strojového učenia s praktickými aspektami prediktívneho riadenia.

Literatúra

[ÅB00] K.J. Åström and R. D. Bell. *Drum-boiler dynamics*, volume 36. Pergamon Press, Inc., Tarrytown, NY, USA, March 2000.

[ÅE72] K. J. Åström and K. Eklund. A simplified non-linear model of a drum boiler-turbine unit. *International Journal of Control*, 16(1):145–169, 1972.

[BBM02] A. Bemporad, F. Borrelli, and M. Morari. Model predictive control based on linear programming - the explicit solution. *IEEE Transactions on Automatic Control*, 47(12):1974–1985, 2002.

[BMDP02] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3 – 20, 2002.

[CR79] C. R. Cutler and B. L. Ramaker. Dynamic Matrix Control. In *AIChE 86th National Meeting*, Houston, Texas, 1979.

[DJKK16] J. Drgoňa, F. Janeček, M. Klaučo, and M. Kvasnica. Regionless explicit mpc of a distillation column. In *European Control Conference 2016*, pages 1568–1573, Aalborg, Denmark, 2016.

[DKJK17] J. Drgoňa, M. Klaučo, F. Janeček, and M. Kvasnica. Optimal control of a laboratory binary distillation column via regionless explicit mpc. *Computers & Chemical Engineering*, 96:139–148, 2017.

[DKK15] J. Drgoňa, M. Klaučo, and M. Kvasnica. Mpc-based reference governors for thermostatically controlled residential buildings. In *54th IEEE Conference on Decision and Control*, volume 54, Osaka, Japan, December 15-18, 2015 2015.

[FBD08] H. Ferreau, H. Bock, and M. Diehl. An online active set strategy to overcome the limitations of explicit mpc. *International Journal of Robust and Nonlinear Control*, 18(8):816–830, 2008.

[Fle13] R. Fletcher. *Practical Methods of Optimization*. Wiley, 2013.

[GBN11] Arun Gupta, Sharad Bhartiya, and P.S.V. Nataraj. A novel approach to multiparametric quadratic programming. *Automatica*, 47(9):2112–2117, 2011.

[GK02] Elmer Gilbert and Ilya Kolmanovsky. Nonlinear tracking control in the presence of state and control constraints: a generalized reference governor. *Automatica*, 38(12):2063 – 2073, 2002.

[GNN18] Emanuele Garone, Marco Nicotra, and Lorenzo Ntogramatzidis. Explicit reference governor for linear systems. *International Journal of Control*, 91(6):1415–1430, 2018.

[HKD+18] J. Holaza, M. Klaučo, J. Drgoňa, J. Oravec, M. Kvasnica, and M. Fikar. Mpc-based reference governor control of a continuous stirred-tank reactor. *Computers & Chemical Engineering*, 108:289–299, 4 january 2018.

[HKK17] J. Holaza, M. Klaučo, and M. Kvasnica. Solution techniques for multi-layer mpc-based control strategies. In *Preprints of the 20th IFAC World Congress, Toulouse, France*, volume 20, July 9-14, 2017 2017.

[HVHDG19] Mehdi Hosseinzadeh, Klaske Van Heusden, Guy A. Dumont, and Emanuele Garone. An explicit reference governor scheme for closed-loop anesthesia. In *2019 18th European Control Conference (ECC)*, pages 1294–1299, 2019.

[KF11] M. Kögel and R. Findeisen. A fast gradient method for embedded linear predictive control. *IFAC Proceedings Volumes*, 44(1):1362–1367, 2011.

[KK17] M. Klaučo and M. Kvasnica. Control of a boiler-turbine unit using mpc-based reference governors. *Applied Thermal Engineering*, 110:1437–1447, 2017.

[KK18] M. Klaučo and M. Kvasnica. Towards on-line tunable explicit mpc using interpolation. In *Preprints of the 6th IFAC Conference on Nonlinear Model Predictive Control*, Madison, Wisconsin, USA, 19-22.8 2018. IFAC.

[KK19a] K. Kiš and M. Klaučo. Neural networks trained as explicit tunable mpc feedback controllers. In M. Fikar and M. Kvasnica, editors, *Proceedings of the 22nd International Conference on Process Control*, Štrbské Pleso, Slovakia, June 11-14, 2019 2019. Slovak University of Technology in Bratislava, Slovak Chemical Library.

[KK19b] M. Klaučo and M. Kvasnica. *MPC-Based Reference Governors*. Springer, 1 edition, 2019.

[KKK15] M. Kalúz, M. Klaučo, and M. Kvasnica. Real-time implementation of a reference governor on the arduino microcontroller. In M. Fikar and M. Kvasnica, editors, *Proceedings of the 20th International Conference on Process Control*, pages 350–356, Štrbské Pleso, Slovakia, June 9-12, 2015 2015. Slovak University of Technology in Bratislava, Slovak Chemical Library.

[KKK17] M. Klaučo, M. Kalúz, and M. Kvasnica. Real-time implementation of an explicit mpc-based reference governor for control of a magnetic levitation system. *Control Engineering Practice*, (60):99–105, March 2017 2017.

[KKK19] M. Klaučo, M. Kalúz, and M. Kvasnica. Machine learning-based warm starting of active set methods in embedded model predictive control. *Engineering Applications of Artificial Intelligence*, 77:1–8, 2019.

[LGKG22] Nan Li, Sijia Geng, Ilya V. Kolmanovsky, and Anouck Girard. An explicit reference governor for linear sampled-data systems with disturbance inputs and uncertain time delays. *IEEE Transactions on Automatic Control*, pages 1–1, 2022.

[LK18] Sergio Lucia and Benjamin Karg. A deep learning-based approach to robust nonlinear model predictive control. *IFAC-PapersOnLine*, 51(20):511–516, 2018. 6th IFAC Conference on Nonlinear Model Predictive Control NMPC 2018.

[LKFM20] Y. Lohr, M. Klaučo, M. Fikar, and M. Mönnigmann. Machine learning assisted solutions of mixed integer mpc on embedded platforms. In *Preprints of the 21st IFAC World Congress (Virtual), Berlin, Germany, July 12-17, 2020*, volume 21, July 12-17, 2020 2020.

[LKKM19] Y. Lohr, M. Klaučo, M. Kalúz, and M. Mönnigmann. Mimicking predictive control with neural networks in domestic heating systems. In M. Fikar and M. Kvasnica, editors, *Proceedings of the 22nd International Conference on Process Control*, pages 19–24, Štrbské Pleso, Slovakia, June 11-14, 2019 2019. Slovak University of Technology in Bratislava, Slovak Chemical Library.

[LLR+19] Kaiwen Liu, Nan Li, Denise Rizzo, Emanuele Garone, Ilya Kolmanovsky, and Anouck Girard. Model-free learning to avoid constraint violations: An explicit reference governor approach. In *2019 American Control Conference (ACC)*, pages 934–940, 2019.

[May14] D. Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967 – 2986, 2014.

[MRRS00] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789 – 814, 2000.

[NW06] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.

[OK22] J. Oravec and M. Klaučo. Real-time tunable approximated explicit mpc (accepted on 24-feb-2022). *Automatica*, 2022.

[PB14] P. Patrinos and A. Bemporad. An accelerated dual gradient-projection algorithm for embedded linear model predictive control. *IEEE Transactions on Automatic Control*, 59(1):18–33, 2014.

[PR03] G. Pannocchia and J. B. Rawlings. Disturbance models for offset-free model-predictive control. *AIChE Journal*, 49(2), February 2003.

[QB03] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733 – 764, 2003.

[RJM12] S. Richter, C. N. Jones, and M. Morari. Computational complexity certification for real-time mpc with input constraints based on the fast gradient method. *IEEE Transactions on Automatic Control*, 57(6):1391–1403, 2012.

[SVR+10] Brett T. Stewart, Aswin N. Venkat, James B. Rawlings, Stephen J. Wright, and Gabriele Pannocchia. Cooperative distributed model predictive control. *Systems & Control Letters*, 59(8):460 – 469, 2010.

[TSG+20] J. Theunissen, A. Sorniotti, P. Gruber, S. Fallah, M. Ricco, M. Kvasnica, and M. Dhaens. Regionless explicit model predictive control of active suspension systems with preview. *IEEE Transactions on Industrial Electronics*, 67(6):4877–4888, 2020.

[WGBM03] W. Wojsznis, J. Gudaz, T. Blevins, and A. Mehta. Practical approach to tuning MPC. *ISA Transactions*, 42(1):149–162, 2003.

[Wil93] H.P. Williams. *Model Building in Mathematical Programming.* John Wiley & Sons, Third Edition, 1993.

[WN10] A. Wills and B. Ninness. Qpc–quadratic programming in c. *Software available at http://sigpromu.org/quadprog*, 2010.

Dodatok A

Originálne práce autora

Nižšie uvedené časopisecké vedecké práce tvoria základné piliere habilitačnej práce autora:

1. **M. Klaučo** – M. Kalúz – M. Kvasnica: Real-time implementation of an explicit MPC-based reference governor for control of a magnetic levitation system. **Control Engineering Practice**, č. 60, str. 99-105, 2017.
2. **M. Klaučo** – M. Kvasnica: Control of a boiler-turbine unit using MPC-based reference governors. **Applied Thermal Engineering**, zv. 110, str. 1437-1447, 2017.
3. J. Holaza – **M. Klaučo** – J. Drgoňa – J. Oravec – M. Kvasnica – M. Fikar: MPC-Based Reference Governor Control of a Continuous Stirred-Tank Reactor. **Computers & Chemical Engineering**, zv. 108, str. 289-299, 2018.
4. **M. Klaučo** – M. Kalúz – M. Kvasnica: Machine learning-based warm starting of active set methods in embedded model predictive control. **Engineering Applications of Artificial Intelligence**, zv. 77, str. 1-8, 2019.
5. Oravec, J. – **M. Klaučo**, M.: Real-Time Tunable Approximated Explicit MPC, **Automatica**, ISSN: 0005-1098, publikácia oficiálne prijatá dňa 24.2.2022.

Prílohu habilitačnej práce tvorí aj úvod a obsah monografie:

1. **M. Klaučo** – M. Kvasnica: MPC-Based Reference Governors, Editor(i): M. J. Grimble, A. Ferrara, Springer, 2019.

Okrem vyššie spomenutých prác patrí k portfóliu autora aj monografia vo vydavateľstve Springer, ktorú z rozsahových dôvodov nie je možné priložiť. Autor počas svojho vedeckého bádania spolupracoval aj na témach ako "optimálne plánovanie trasy" alebo "znižovanie zložitosti explicitného MPC", z ktorých spolu-publikoval ďalšie **3 karentované články**.

Celkový publikačný profil autora sa nachádza na webových sídlach:

- Domovská stránka: https://www.uiam.sk/~klauco/
- Web of Science profil: https://publons.com/researcher/2400672/martin-klauco/
- Google Scholar profil: https://scholar.google.com/citations?hl=sk&user=wVXDzr8AAAAJ
- ORCID: https://orcid.org/0000-0003-0098-2625

Control Engineering Practice 60 (2017) 99–105

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Real-time implementation of an explicit MPC-based reference governor for control of a magnetic levitation system

Martin Klaučo*, Martin Kalúz, Michal Kvasnica

Slovak University of Technology in Bratislava, Radlinského 9, SK-812 37 Bratislava, Slovak Republic

ARTICLE INFO

Keywords:
Explicit MPC
Reference governor
Magnetic levitation
Real-time implementation

ABSTRACT

This paper deals with the design and real-time implementation of an Model Predictive Control (MPC)-based reference governor on an industrial-like microcontroller. The task of the governor is to provide optimal setpoints for an inner Proportional-Summation-Difference (PSD) controller. The MPC-based governor is synthesized off-line as a Piecewise Affine (PWA) function that maps measurements onto optimal references. To achieve a fast and memory-efficient implementation, the PWA function is encoded as a binary search tree. This allows the reference governor to run on a sub-millisecond scale even on a very simple hardware. The proposed concept is experimentally verified on a laboratory device involving a magnetic levitation system. Here, the PSD controller is responsible for controlling the vertical position of the ball in the magnetic field. By using the reference governor, control performance can be significantly improved and input/output constraints enforced in a systematic manner.

1. Introduction

The vast majority of industrial automation relies on the use of simple closed-loop control approaches such as the PID control (Åström & Hägglund, 2006). These controllers are very popular due to their simplicity and ability to provide analytically determinable quality of closed-loop behavior. One of the main drawbacks, however, is their inability to explicitly cope with constraints on inputs and outputs. When constraint satisfaction is of imminent importance, MPC (Maciejowski, 2002) is usually employed. MPC is a control strategy which is based on determining optimal control actions by solving an optimal control problem at each sampling instant. Despite it being able to cope with constraints in a systematic manner, the computational load due to repetitive optimization only makes it suitable for systems with slow dynamics. This limitation can be abolished, to a certain extent, by shifting the optimization off-line by using the concept of explicit MPC (Bemporad, Morari, Dua, & Pistikopoulos, 2002). Here, parametric optimization is used to pre-calculate the optimal feedback law in the form of a look-up table. This allows the concept of MPC to be applied to fast systems as well.

However, the transition from PID controllers to MPC strategies is often not smooth, especially if significant effort was already invested towards tuning of the PID controller(s). Simply put, it is often preferred to keep existing PID controllers in place, and use MPC-like optimization on top of them to improve performance and plant safety. This strategy, known as the Reference Governor (RG), is well known in the literature, see, e.g., Bemporad (1998), Borrelli, Falcone, Pekar, and Stewart (2009), and Gilbert and Kolmanovsky (1999). The task of the governor is to shape the user-specified desired reference in such a way that input/output constraints are enforced and closed-loop performance is improved. The main drawback of existing solutions lies in their underlying computational complexity. Since the shaping is performed by solving an optimization problem at each sampling instant, significant processing power is typically required. This is an even bigger issue if nonlinear MPC is considered. In such a case, a requirement on constraints satisfaction is often omitted or reduced (e.g. only input constraints are considered), in order to push the complexity to a realizable scale (Bächle, Hentzelt, & Graichen, 2013).

In this work, it will be shown how to design and implement a reference governor on a simple and cheap microprocessor in real time. A magnetic levitation system is considered for demonstrating control performance. Such systems have a broad use in industrial applications where they serve to suspend materials in a magnetic field. For instance, it is already used in maglev trains (Lee, Kim, & Lee, 2006), high-speed motors using magnetic bearings (Schuhmann, Hofmann, & Werner, 2012), but also in some unusual applications such as 3D cell culturing (Haisler et al., 2013) or harvesting of kinetic energy from human movements (Berdy, Valentino, & Peroulis, 2015).

Various control strategies for control of magnetic levitation are available in the literature. In Glück, Kemmetmüller, Tump, and Kugi

* Corresponding author.
E-mail addresses: martin.klauco@stuba.sk (M. Klaučo), martin.kaluz@stuba.sk (M. Kalúz), michal.kvasnica@stuba.sk (M. Kvasnica).

http://dx.doi.org/10.1016/j.conengprac.2017.01.001
Received 12 August 2016; Received in revised form 30 November 2016; Accepted 4 January 2017

(2011), a cascaded controller for self-sensing magnetic levitation along with the position estimation based on least squares identification, implemented on Altera Stratix II Field Programmable Gate Array (FPGA), is proposed. A wholly different approach to magnetic levitation control is provided in Bächle et al. (2013), where a computationally low-demand nonlinear MPC is demonstrated on dSPACE platform. A robust tracking control of magnetic levitation process with input constraints is proposed in Zhang, Xian, and Ma (2015). Here the authors use the MATLAB-equipped PC to control the system. An interesting approach of magnetic levitation control design is provided in Folea, Muresan, Keyser, and Ionescu (2016), where a fraction order controller implemented in embedded microcontroller is used to increase the closed-loop performance and robustness. Very popular approach to the application of control algorithms for magnetic processes is the use of FPGAs (Lin, Lin, & Chen, 2011; Lin, Liu, & Li, 2014).

In this paper, a parametric solution to the RG problem will be derived, and it will be used to optimally control the closed-loop system with magnetic levitation. This is achieved by solving the optimization problem using parametric optimization (Bemporad et al., 2002; Gaas & Saaty, 1955), which yields the analytical solution in a form of a look-up table. Then computation of the optimally shaped references reduces to a simple table lookup, which can be performed very quickly even on a simple implementation hardware. To demonstrate this advantage of explicit MPC-based RG a real control experiment with low-cost microcontroller unit (MCU) SAM3X (based on Cortex-M3) has been performed. This MCU provides memory and performance characteristics in orders of magnitude lower compared to processors that are usually used for implementation of online MPC.

Previous work of Kalúz, Klaučo, and Kvasnica (2015) has shown a design of an MPC RG for the system of magnetic levitation, based on the input/output model. The controller was implemented on very limiting hardware, namely the Atmel ATMega328p 8bit microcontroller with only 2 kB of dynamic memory. This limited the MPC formulation in terms of closed-loop model complexity, prediction horizon, and constraints that could be defined only for a position of the levitating object. Specifically, only output constraints were handled in the previous work and the constraints on the control action generated by the PID controller were not enforced. This was a consequence of using an input–output prediction model.

In this work, the prior art is improved in several key respects. First, a state-space model of the closed-loop system is considered. This allows not only to explicitly account for all types of constraints (e.g., constraints on the control inputs as well as on controlled outputs), but also to improve the control performance. Although tailored to the specific application of magnetic levitation, this paper proposes a unified mathematical/software framework for the design and implementation of MPC-based reference governors. This framework is applicable to any closed-loop systems that can be represented by a linear state-space model. Moreover, to reduce the on-line implementation effort, the concept of binary search trees (Tøndel, Johansen, & Bemporad, 2003) is used allowing for efficiently encoding the parametric representation of the reference governor.

The proposed parametric reference governor setup is applied to a fast and unstable laboratory device of magnetic levitation. Despite the fact that the MCU provides only modest computational resources and a small amount of memory storage (96 kB), it can accommodate the parametric reference governor and execute it under hard real-time bounds. Experimental results are presented that confirm the efficiency of the proposed solution. The experiment is implemented on small-scale laboratory magnetic levitation device that provides modest parameters and limited dimensions. The used control hardware (Atmel SAM3X) is even more limiting and proves the memory and computational efficiency of proposed control scheme. The comparison of presented solution similar to the above-mentioned works by other authors is provided in Section 4.7.

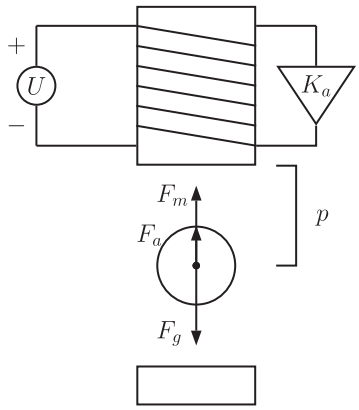


Fig. 1. Magnetic levitation system.

2. Mathematical model of magnetic levitation

The mathematical model of magnetic levitation is derived from the standard physical understanding of magnetic suspension of an object. The spatial arrangement of the system is shown in Fig. 1. The arrangement consists of three main parts. These are a metal cylinder with a winding that makes an electromagnetic coil at the top, inductive proximity sensor at the base, and ferromagnetic ball in the space between them. The arrangement considers the displacement of the ball only in the vertical axis (denoted as p).

The dynamics of the ball movement is derived from a standard force equation $F_\mathrm{a} = F_g - F_\mathrm{m}$, where $F_\mathrm{a} = m_\mathrm{b}\frac{d^2p}{dt^2}$ is the acceleration force, $F_g = m_\mathrm{b}g$ is the gravitational force, and $F_\mathrm{m} = K_\mathrm{m}\frac{I_\mathrm{c}^2}{p^2}$ is the force of magnetic field acting on the ball. Since the practical implementation relies on inputs and outputs to be in voltage units, the input voltage applied to the coil, i.e., U_c, can be expressed as $U_\mathrm{c} = I_\mathrm{c}R_\mathrm{c}$. Similarly, the output voltage taken from the position sensor, i.e., U_ps, can be expressed as a linear function of position $U_\mathrm{ps} = a_\mathrm{ps}p + b_\mathrm{ps}$. Using these two substitutions and introducing a conjured coil amplifier constant $K_\mathrm{a} = \frac{K_\mathrm{m}}{R_\mathrm{c}^2}$, the model can be written as

$$\frac{m_\mathrm{b}}{a_\mathrm{ps}}\frac{d^2U_\mathrm{ps}}{dt^2} = m_\mathrm{b}g - K_\mathrm{a}\frac{U_\mathrm{c}^2 a_\mathrm{ps}^2}{U_\mathrm{ps}^2 - 2b_\mathrm{ps}U_\mathrm{ps} + b_\mathrm{ps}^2}. \tag{1}$$

This model, however, is nonlinear in U_c and U_ps. Therefore a linearization around the equilibrium U_c^s and U_ps^s where $F_\mathrm{a} = 0$ was considered. Applying first-order Taylor expansion to (1) yields a linear dynamics of the form

$$\frac{d^2y}{dt^2} = k_1 y + k_2 u, \tag{2}$$

where $y = U_\mathrm{ps} - U_\mathrm{ps}^\mathrm{s}$ and $u = U_\mathrm{c} - U_\mathrm{c}^\mathrm{s}$ are the deviations of output and input voltages from the selected equilibrium point, respectively. The output deviation of the system y is in the unit of volts and it linearly represents the actual physical distance p. The constants k_1 and k_2 are given by

$$k_1 = 2K_\mathrm{a}\frac{a_\mathrm{ps}^3 U_\mathrm{c}^{\mathrm{s}2}}{m_\mathrm{b}(U_\mathrm{ps}^\mathrm{s} - b_\mathrm{ps})^3}, \quad k_2 = -2K_\mathrm{a}\frac{a_\mathrm{ps}^3 U_\mathrm{c}^\mathrm{s}}{m_\mathrm{b}(U_\mathrm{ps}^\mathrm{s} - b_\mathrm{ps})^2}, \tag{3}$$

which, using the values from Table 1, amounts to $k_1 = 2250$, $k_2 = 9538$. The system in (2) is subsequently converted into the state-space form

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ k_1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ k_2 \end{bmatrix} u(t), \tag{4a}$$

$$y(t) = [1 \;\; 0]x(t), \tag{4b}$$

and discretized using a sampling time T_s. The PID controller used in

Table 1
Parameters of the magnetic levitation plant.

Variable	Unit	Value
m_b	kg	$8.4{\cdot}10^{-3}$
g	m s^{-2}	9.81
a_{ps}	V m^{-1}	$-1.1132{\cdot}10^{3}$
b_{ps}	V	11.107
U_c^s	V	2.29
U_{ps}^s	V	1.40
K_a	kg m^2 s^{-2} V^{-1}	$1.1899{\cdot}10^{-6}$

this work is in the form of

$$G_{\mathrm{PID}}(s) = \frac{13.07s^2 + 1358s + 4.367{\cdot}10^4}{s^2 + 571.4s}. \tag{5}$$

3. Synthesis of optimization-based reference governors

In this section, a formulation of the MPC-based reference governor will be shown. First, the state-space model of the closed-loop system is derived, followed by formulation of the reference governor as an MPC controller. The MPC problem is subsequently solved parametrically as to obtain the explicit analytical representation of the reference governor.

3.1. Closed-loop modeling

The control setup considered in this paper is depicted in Fig. 2 and it consists of the inner-loop PSD controller and the outer-loop reference governor that optimizes the setpoints $w(t)$ for the PSD controller. The reference governor, presented in the next section, employs a state-space representation of the inner feedback loop that can be compactly represented by the extended state-space model of the form

$$\tilde{x}(t + T_s) = \widetilde{A}\tilde{x}(t) + \widetilde{B}w(t), \tag{6a}$$

$$u(t) = \widetilde{C}_u\tilde{x}(t) + \widetilde{D}_u w(t), \tag{6b}$$

$$y(t) = \widetilde{C}_y\tilde{x}(t) + \widetilde{D}_y w(t), \tag{6c}$$

where the extended state vector $\tilde{x}$

$$\tilde{x} = [x_r^\top \;\; x^\top]^\top \tag{7}$$

consists of the states x of the controlled plant in (4), as well as of the states x_r of the inner-loop PSD controller. At this point, a discrete-time version of the controller in (5) is considered which, after conversion to the state-space form, can be represented by

$$x_r(t + T_s) = A_r x_r(t) + B_r e(t), \tag{8a}$$

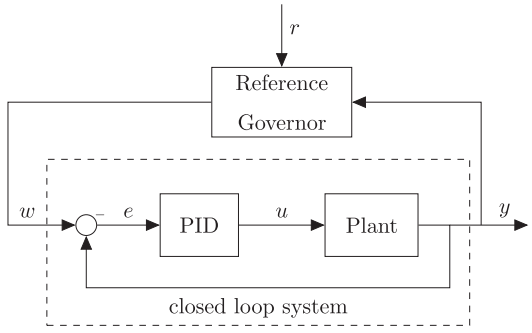


Fig. 2. Scheme of the reference governor setup.

$$u(t) = C_r x_r(t) + D_r e(t). \tag{8b}$$

Here, $e(t) = w(t) - y(t)$ represents the control error. Then the matrices of an extended state-space model in (6) can be derived using straightforward algebraic manipulation that is discussed in detail in Klaučo (2016).

3.2. Synthesis of a reference governor

The MPC-based reference governor that optimizes the setpoints w of the inner feedback loop in (6) is formulated as a constrained finite time-optimal control problem of the form

$$\min_{w_0,\ldots,w_{N-1}} \sum_{k=0}^{N-1} \Big(\| y_k - r \|^2_{Q_y} + \| w_k - w_{k-1} \|^2_{Q_w} + \| u_k - u_{k-1} \|^2_{Q_u} + \\ + \| w_k - r \|^2_{Q_{wr}} + \| y_k - w_k \|^2_{Q_{yw}} \Big) \tag{9a}$$

$$\text{s. t.} \quad \tilde{x}_{k+1} = \widetilde{A}\tilde{x}_k + \widetilde{B}w_k, \;\; \forall\, k \in \mathbb{N}_0^{N-1}, \tag{9b}$$

$$u_k = \widetilde{C}_u\tilde{x}_k + \widetilde{D}_u w_k, \qquad \forall\, k \in \mathbb{N}_0^{N-1}, \tag{9c}$$

$$y_k = \widetilde{C}_y\tilde{x}_k + \widetilde{D}_y w_k, \qquad \forall\, k \in \mathbb{N}_0^{N-1}, \tag{9d}$$

$$u_{\min} \le u_k \le u_{\max}, \qquad \forall\, k \in \mathbb{N}_0^{N-1}, \tag{9e}$$

$$y_{\min} \le y_k \le y_{\max}, \qquad \forall\, k \in \mathbb{N}_0^{N-1}, \tag{9f}$$

$$w_k = w_{N_c-1}, \qquad \forall\, k \in \mathbb{N}_{N_c}^{N-1}, \tag{9g}$$

where $\tilde{x}_k$, u_k, y_k represent the k-th step predictions of the inner loop's states, control inputs of the inner controller, and plant outputs, respectively. Next, (9a) is a quadratic cost function with prediction horizon N. The cost function (9a) penalizes weighted squared 2-norms of respective quantities with $\| z \|^2_M = z^\top M z$, $M \succeq 0$. The first term of the cost function minimizes the tracking error and forces the plants output, in this case the output voltage, to track a user defined reference r. The second and the third term penalize fluctuations of the optimized setpoints w and the control actions of the primary PSD controller, respectively. The fourth term accounts for the difference between reference r and shaped setpoint w and finally the fifth term penalizes the deviations between predicted output and shaped setpoint for the inner controller. The tuning factors Q_y, Q_w, Q_u, Q_{wr} and Q_{yw} are positive definite matrices of suitable dimensions. The PSD controller output u and the plants output y are linked via constraints (9c) and (9d) with the optimized setpoint w. The constraints in (9e) ensure that the chosen setpoint profile $W = [w_0^\top, \ldots, w_{N-1}^\top]^\top$ satisfies physical limits on the control inputs u supplied by the PSD controller. Finally, (9g) represents a move blocking constraint which employs the control horizon $N_c \le N$ and is used to decrease the number of degrees of freedom and thus makes the problem simpler to solve.

The optimization problem in (9) is initialized by the vector of parameters

$$\theta = [x_r(t)^\top \;\; x(t)^\top \;\; w(t - T_s)^\top \;\; u(t - T_s)^\top \;\; r(t)^\top]^\top, \tag{10}$$

where $x_r(t)$ are the states of the inner PSD controller at time t and $x(t)$ are the states of the plant. Moreover, $w(t - T_s)$ is the value of the optimized reference at the previous sampling instant, required in (9a) at $k = 0$ when $w_{-1} = w(t - T_s)$. Similarly, $u(t - T_s)$ is the control input generated by the PSD controller at the previous sampling instant that is used as u_{-1} in (9a). Finally, $r(t)$ is the user-defined reference. Because of (4), the plant's states $x(t)$ consist of the ball's position $y(t)$ and its speed. Since only the position can be directly measured, the ball's position is estimated by $\frac{y(t) - y(t - T_s)}{T_s}$.

With the initial conditions as in (10), the optimal modulated references $w_0^\star, \ldots, w_{N-1}^\star$ can be obtained by solving (9) as a quadratic program (QP). To allow for a fast and efficient implementation of the solution on an embedded hardware, it is furthermore proposed to

obtain an explicit solution to the optimization problem (9).

3.3. Parametric solution to (9)

It is well known (see, e.g., Bemporad et al., 2002) that the MPC optimization problem in (9) can be re-cast as a convex parametric quadratic program (pQP) whose analytic solution takes the form of a continuous piecewise affine (PWA) function

$$W^\star(\theta) = \begin{cases} \alpha_1\theta + \beta_1 & \text{if } \theta \in \mathcal{R}_1 \\ \vdots & \\ \alpha_L\theta + \beta_L & \text{if } \theta \in \mathcal{R}_L \end{cases} \tag{11}$$

Here, $\mathcal{R}_i = \{\theta \mid Z_i\theta \le z_i\}$ for $i = 1,\dots,L$ are polyhedral critical regions, L denotes the total number of regions and α_i, β_i are the coefficients of the i-th locally optimal feedback law. The critical regions, as well as the local affine feedback laws, are constructed by applying parametric programming solvers (Bemporad et al., 2002; Gupta et al., 2011; Pistikopoulos et al., 2002), which are available e.g. in the MPT Toolbox (Herceg, Kvasnica, Jones, & Morari, 2013) or in the POP toolbox (Pistikopoulos et al., 2015). It is important to note that the explicit optimizer $W^\star(\theta)$ as in (11) is constructed off-line.

Once the control law $W(\theta)$ in a form of a PWA function is known, the calculation of the actual set point $w^\star(t)$ for the PSD controller then reduces to a mere evaluation of the PWA function in (11). Note that the value of the optimally shaped setpoint profile $w_0^\star,\dots,w_{N-1}^\star$ obtained via evaluating (11) is the same as a numerical solution to the QP (9) for a given value of initial condition θ, see Bemporad et al. (2002) for a proof.

The most efficient way of evaluating the function $W^\star(\theta)$ for a particular value of the vector of parameters θ is to organize the critical regions $\mathcal{R}_i$ into a binary search tree (Tøndel et al., 2003), which is schematically depicted in Fig. 3. In each node of the binary search tree, the critical regions $\mathcal{R}_i$ are split by a suitable separating hyperplane into two parts of roughly identical cardinality. Selection of the most suitable separating hyperplane is based on the so-called *hyperplane arrangement* (Geyer, Torrisi, & Morari, 2008) of critical regions. The arrangement determines on which side of each candidate hyperplane each critical region is contained. The candidates are selected as the unique hyperplanes that determine each critical region $\mathcal{R}_i$ and therefore up to $O(L^2)$ operations are required to construct the hyperplane arrangement. With the arrangement in hand, the best separating hyperplane in each node is selected as the one that minimizes the maximal number of regions on each side of the hyperplane.

An algorithm that computes the hyperplane arrangement and constructs the binary search tree is available in the MPT toolbox. Once constructed, the binary tree allows to evaluate the function $W^\star(\theta)$ in time that is logarithmic in the number of critical regions.

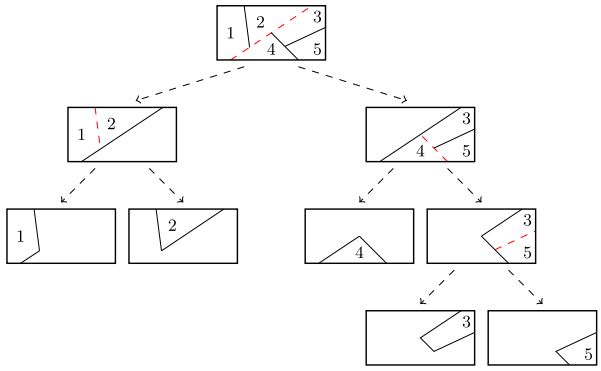


Fig. 3. Regions organized into a binary search tree. Red dashed lines denote the separating hyperplanes. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

3.4. Feedback implementation

The feedback implementation of the parametric form of the reference governor is based on obtaining the optimally shaped references $w_0^\star,\dots,w_{N-1}^\star$ by evaluating the PWA function $W^\star(\theta)$. At each time t, one first forms the vector of parameters θ per (10). Subsequently, the PWA function (11) is evaluated for the given value of θ by traversing the associated binary search tree. In the spirit of receding horizon implementation (Mayne, Rawlings, Rao, & Scokaert, 2000), only the first element of $W^\star$, i.e., $w_0^\star$, is implemented as the reference for the inner PSD controller. The whole procedure is then repeated from the beginning at the subsequent time instant *ad infinitum*.

4. Implementation

4.1. Laboratory magnetic levitation plant

The actual physical system of magnetic levitation used in this work is the CE152 laboratory model, shown in Fig. 4. The plant consists of an electromagnetic coil energized by a power amplifier, a linear position sensor and a ferromagnetic ball. The intensity of magnetic field is a manipulated variable and can be directly influenced by voltage electric signal in the range of 0–5 V. A measured variable is the reading of the proximity position sensor in the same 0–5 V range.

This system exhibits several properties that result in interesting control design challenges. One of them is the fast dynamics of the system. The time constant of the open loop is approximately 20 ms, which indicates that reasonable sampling rate should be chosen not less than 0.5 kHz to get high-quality measurements. In order to demonstrate the time efficiency of explicit MPC, an even higher sampling rate of 1 kHz has been chosen in this work. Other challenging properties of the system are the natural instability and nonlinearity. The instability comes from the spatial arrangement of laboratory model, where the levitating object is pulled upwards to the magnetic coil against the force of the gravity. This is the exact opposite of naturally stable magnetic repulsion arrangement used, e.g., by the *maglev* train. Moreover, the system also exhibits a strong nonlinear behavior. As it is obvious from the force balance equation, the force applied on ball F_a is a sum of gravitational force F_g and force of the magnetic field F_m. While F_g can be considered constant with always present effect, the F_m is a quadratic function of manipulated variable (input voltage U_c) and is present only when nonzero voltage is applied. This results into asymmetric nonlinearity, depending on the direction of ball position control.

4.2. Microcontroller

Electronic controller used in this work is an embedded MCU Atmel SAM3X8E. This MCU is based on the 32-bit ARM Cortex-M3 processor, supported by 84 MHz clock, 512 kB of flash memory, and 96 kB of SRAM in two banks (32 kB and 64 kB). As stated by Atmel, a SAM3X series is designed for the applications in the industrial embedded market, home and building automation, small grids, industrial automation and networking. This is one of the reasons why this MCU was used in this work as a physical controller. The second reason is the modest computational and memory capabilities of the device. Specifically, they allow to stand out for the low implementation complexity of the explicit form of the solution, as given per (11).

4.3. Experiment setup

The magnetic levitation system CE152 directly provides a connector with raw analog electrical signals for control and data acquisition. Both, the signals for control of coil amplifier and position sensor, are voltage signals in the range of 0–5 V. One of the implementation issues was the interconnection of a physical system with the MCU that operates on

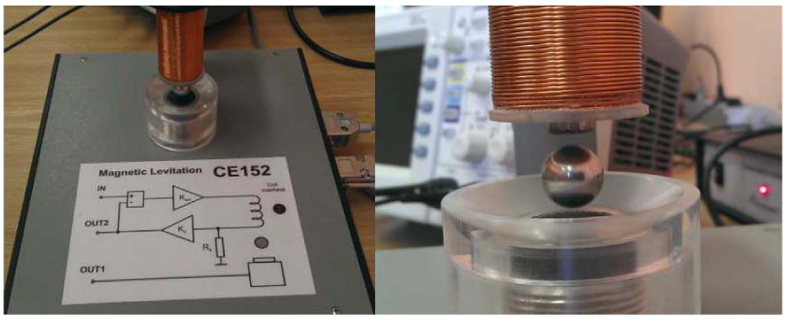


Fig. 4. Laboratory model of magnetic levitation CE152.

3.3 V. This problem was solved by using a signal level conversion. For connection of position sensor signal to the MCU, an integrated 12bit Analog-to-Digital Converter (ADC) of SAM3X was used. To scale the signal to a lower range, a voltage divider was used. The control signal (coil voltage) is issued by an external serial 12bit Digital-to-Analog Converter (DAC). This DAC is controlled via a Serial Peripheral Interface (SPI). All source and control lines for DAC have been scaled through the logic level converter from 3.3 V to 5 V. This physical setup provides satisfactory precision and performance parameters for the desired application.

4.4. Software implementation of the RG-MPC controller

The reference governor in a form of an MPC problem (9) was constructed using YALMIP (Löfberg, 2004). The prediction horizon N set to 9 samples and the control horizon $N_c = 2$ was used. Because of physical constraints of the plant, the control action u was limited by $u_{min} = 0.79$ V, $u_{max} = 4.79$ V, along with constraints on the ball's position y_{min}=1.00 V, y_{max}=2.00 V. The voltage constraints on ball's position directly translate to the physical limits of $p \in [8.18,\ 9.08]$ mm, cf. Section 2. The weighting factors in (9a) were chosen as $Q_y = 10^{-2}$, $Q_w = 10^{-3}$, $Q_u = 1$ $Q_{wr} = 0.1$, $Q_{yw} = 0.1$.

State space matrices of the closed loop system were discretized with sampling time $T_s = 1$ ms, resulting in

$$\widetilde{A} = \begin{bmatrix} 0.6749 & -0.0355 & -0.0008 & -0.0000 \\ 0.0008 & 1.0000 & -0.0000 & -0.0000 \\ -8.8304 & 49.0588 & 0.9727 & 0.0010 \\ -16470.9632 & 97772.4837 & -51.6329 & 0.9751 \end{bmatrix}, \tag{12a}$$

$$\widetilde{B} = \begin{bmatrix} 0.0008 \\ 0.0000 \\ 0.0296 \\ 56.2470 \end{bmatrix}, \tag{12b}$$

$$\widetilde{C}_u = [-2418.9076\ \ 11\,781.9582\ \ -7.8599\ \ 0.0000], \tag{12c}$$

$$\widetilde{D}_u = 7.8599, \tag{12d}$$

$$\widetilde{C}_y = [-5.1447\ \ 25.0588\ \ 1.0000\ \ 0.0000], \tag{12e}$$

$$\widetilde{D}_y = 0. \tag{12f}$$

Next, the PWA representation of the parametric RG optimizer $W^\star(\theta)$ as in (11) was obtained by solving (9) parametrically using the MPT toolbox (Herceg et al., 2013). The parametric solution was obtained in 15 s on a 2.9 GHz Core i7 CPU with 12 GB RAM. The explicit PWA representation of the optimizer $W^\star(\theta)$ in (11) consisted of $L = 21$ critical regions in a 7 dimensional parametric space, cf. (10). The MPT toolbox was subsequently used to synthesize the corresponding binary search tree to speed up the evaluation of the PWA function in (11). The total memory footprint of the binary tree was 1392 bytes. This number includes both the separating hyperplanes and the parameters of local affine feedback laws α_i, β_i associated to each critical region. The construction of the tree took 5 s.

4.5. Closed-loop implementation

The control algorithms are executed in real time. This is ensured by a timer-based program interrupt that fires every 1 ms. This represents the main sampling period T_s. Precise timing of control is carried out by 32bit Timer/Counter (TC) operating in compare mode. The TC interrupt invokes the Interrupt Service Routine (ISR) that has higher priority than the main program, and is served immediately. During the period of T_s, following tasks have to be performed inside the ISR: (i) acquisition of the current ball's position $y(t)$; (ii) construction of the vector of parameters θ via (10); (iii) evaluation of $W^\star(\theta)$ from (11) by traversal of the associated binary search tree; (iv) extraction of $w^\star(t) = w_0^\star$ from $W^\star$; (v) computation of the PSD control action $u(t)$ based on the optimal reference $w^\star(t)$; and (vi) application of $u(t)$ to the system. This loop is then repeated at each sampling instant T_s. The overall control algorithm, which includes the implementation of the PSD controller via (8), the binary search tree, and all necessary auxiliary operations (memory allocations, matrix/vector multiplications, etc.) was written in the C language and subsequently compiled using the ARM GCC compiler. The size of the resulting binary code was 36 152 bytes. Worth noting is that only a tiny fraction of this code pertains to the implementation of the PSD controller and the traversal of the binary search tree. Majority of the code covers auxiliary operations elaborated above. To run the control algorithm, 5180 bytes of dynamic memory were required.

4.6. Data acquisition

The evaluation of experimental results (Section 5) requires a precise measurement of manipulated variable U_c, controlled variable U_{ps}, and also the actual value of shaped and user-defined reference. To measure actual physical values of input and output voltages, these signals have been probed and acquired by a digital oscilloscope. For this purpose a 100 MHz 4-channel oscilloscope was used. The shaped and user-defined references are calculated/defined in the internal algorithm of MCU, and therefore they have been pulled out to the signal interface as electric signals. For the evaluation of the reference change timing, the user-defined reference was hooked to the digital output pin, because it switches only between two different values. This value was used only for the time synchronization of acquired signals. The shaped reference value computed by RG was hooked up to one of the internal 12bit DACs pulled out as a separate analog voltage signal. Both reference signals were then measured along with the input and output signals by an oscilloscope, using the high memory depth settings, to acquire precise measurements.

4.7. Comparison

To highlight the main benefits and drawbacks of the proposed

Table 2
Comparison to other related works.

	Bächle et al. (2013)	Folea et al. (2016)	Lin et al. (2014)	This paper
Control scheme	Nonlinear online MPC	Fractional order contr.	FLCMAC	Explicit MPC
Constraints handling	Input	None	None	Input, output, states
Controller	dSPACE MicroAutoBox I 800 MHz	NI cRIO-9014 Embed. Contr. 400 MHz	Altera Stratix II FPGA	Atmel SAM3X Cortex-M3 84 MHz
Controller price [€]	Thousands	Thousands	Hundreds	Tens
Computation time	900 µs	N/A (<2 ms)	N/A (<100 µs)	266 µs
Sampling frequency	700 Hz	500 Hz	10 kHz	1 kHz

control scheme, a brief comparison to works by Bächle et al. (2013), Folea et al. (2016), and Lin et al. (2014) is provided in Table 2. The main difference lies in the cost of the hardware platforms that were used to implement respective control algorithms. In all referenced works, expensive hardware (dSPACE, embedded controllers, FPGA) was used. Despite the cost and the associated computational power of the alternative approaches, the techniques of this paper outperform, in terms of achievable sampling frequencies, all but one approach. The only exception is Lin et al. (2014) where an FPGA was used. Nevertheless, the Altera Stratix II FPGA is still an order of magnitude more expensive than the Atmel microcontroller used in this paper. More importantly, the competing approaches provide either none or only limited, systematic handling of constraints.

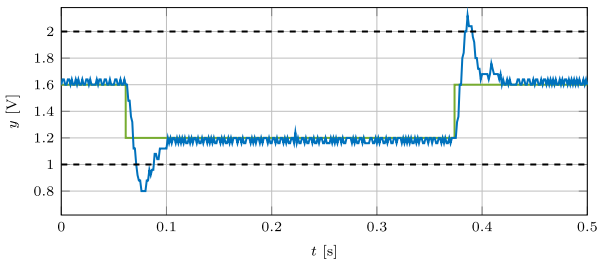
Worth noting is that individual references assume differently sized coils and thus allow for different levitation heights. In this paper, only a small coil is used that provides the levitation height of 0.9 mm, with the maximum height (determined by physical dimensions of the system) of 5 mm. These numbers are smaller than levitation heights in other works (30 mm/70 mm in Bächle et al., 2013, 5 mm/9 mm in Folea et al., 2016, and 10 mm in Lin et al., 2014). However, there is no direct relation between the physical dimensions of the controlled system to the type of employed control algorithm and/or the implementation hardware. The procedures of this paper can be directly applied to control larger coils as well.

5. Experiments and results

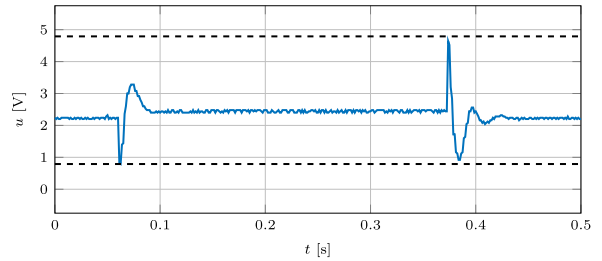
Experimental results demonstrating benefits of the MPC-based reference governor are presented in this section. Two experimental scenarios are considered. In both experiments, the sampling time was chosen as $T_s = 1$ ms. Moreover, the reference was periodically switched between ± 0.2 V (± 0.18 mm) around the chosen equilibrium point $U_{ps}^s = 1.4$ V ($p^s = 8.7$ mm).

The first experiment concerns controlling the vertical position of the ball solely by the PSD controller (8), which is the discretized version of the PID controller stated in (5). In this experiment, constraints on the control input were enforced by artificially saturating the control actions. Naturally, the PSD controller provides no a priori guarantee of satisfying the output constraints. The experimental results are shown in Fig. 5. As expected, the control inputs of the PSD controller shown in Fig. 5(b) obey the limits on the control action due to the saturation. However, the output constraints are violated, as can be seen in Fig. 5(a). Moreover, the control profile exhibits significant under- and overshoots during setpoint changes.

To improve the performance, in the second experiment the MPC-based reference governor was inserted into the loop. The experimental data are visualized in Fig. 6. First and foremost, the reference governor shapes the reference in such a way that constraints on the controlled output (i.e., on the ball's position) are rigorously enforced, cf. Fig. 6(a). This is a consequence of accounting for the constraints directly in the optimization problem, cf. (9f). Moreover, the under- and overshoots are considerably reduced. This shows the potential of MPC-based reference governors to significantly improve the control performance. The improvement is due to the optimal modulation of the user-defined reference as shown in Fig. 6(b). As can be observed, just small



(a) Measurement of the position $y(t)$



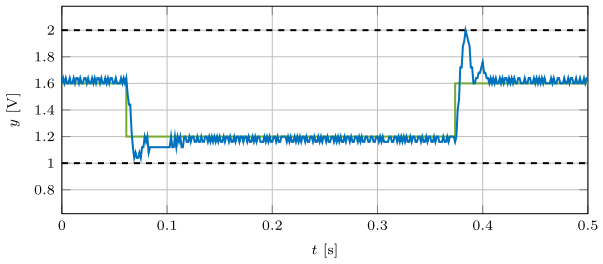
(b) Control action $u(t)$

Fig. 5. Control by the PSD controller only. The top figure shows the user-defined reference $r(t)$ in green and the ball's position $y(t)$ in blue. The bottom figure shows the control action $u(t)$ of the PSD controller, along with saturation input constraints in dashed-black. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

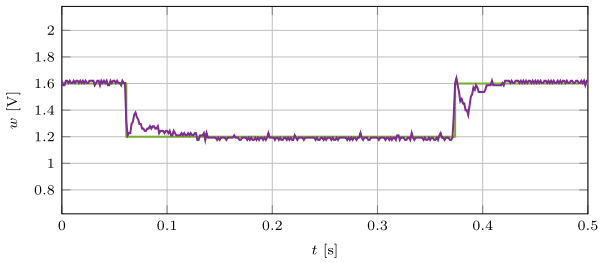
modifications of the reference are required to significantly improve performance. Finally, as can be observed from Fig. 6(c), the reference governor shapes the reference in such a way that the inner PSD controller provides satisfaction of input constraints since they are explicitly embedded in (9e). The evaluation of the RG feedback law (11), encoded as a binary search tree, took 254 µs on average. The best-case evaluation time was 243 µs while the worst case was 266 µs. These fluctuations are caused by the binary search tree not being perfectly balanced, which is an inherent property of the geometric structure of the parametric solution. As a consequence, different numbers of steps are required to evaluate the tree for different values of the parameter θ in (10).

6. Conclusions

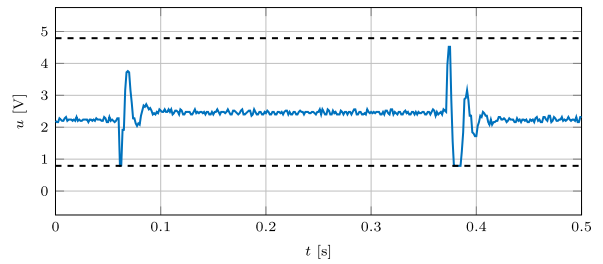
This paper has shown that the performance of a closed-loop system can be significantly improved by optimally modulating the setpoint supplied to the inner controller. The setpoint was optimally shaped using an MPC-based strategy. The MPC-based reference governor is based on a state space representation of the closed-loop model, which allows to account for constraints on the control input generated by the inner controller, as well as on the controlled outputs. To allow for a real-time implementation of the proposed strategy, the MPC-based reference governor problem can be solved off-line using parametric programming. This results in a feedback strategy in the form of a PWA



(a) Measurement of the position $y(t)$



(b) Optimally shaped setpoint $w(t)$



(c) Control action $u(t)$ of the PSD controller

Fig. 6. Control under the reference governor. The top figure shows the measured position of the ball $y(t)$ (blue line), while the reference $r(t)$ is depicted by green. Dashed lines denote output constraints. The middle figure represents the optimally shaped setpoint $w(t)$ (magenta) and the user-defined reference $r(t)$ (green). The bottom figure shows the manipulated variable $u(t)$ and its constraints in dashed-black. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

function. Furthermore, to decrease the on-line computation effort associated with evaluation of such a function, the PWA function can be trans-coded into a binary search tree. By doing so, the reference governor features a low memory footprint of roughly 5 kB. Moreover, since the optimal references are precalculated in the form of a PWA function, the on-line implementation effort is modest as well. Specifically, it never exceeded 266 μs even on a very simple microprocessor.

Acknowledgments

The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grant no. 1/0403/15, and the financial support of the Slovak Research and Development Agency under the project APVV-15-0007.

References

Åström, K. J., & Hägglund, T. (2006). Advanced PID control. ISA — The Instrumentation, Systems, and Automation Society.

Bächle, T., Hentzelt, S., & Graichen, K. (2013). Nonlinear model predictive control of a magnetic levitation system. *Control Engineering Practice*, 21(9), 1250–1258.

Bemporad, A. (1998). Reference governor for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 43(3), 415–419.

Bemporad, A., Morari, M., Dua, V., & Pistikopoulos, E. N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.

Berdy, D., Valentino, D., & Peroulis, D. (2015). Kinetic energy harvesting from human walking and running using a magnetic levitation energy harvester. *Sensors and Actuators A: Physical*, 222, 262–271.

Borrelli, F., Falcone, P., Pekar, J., & Stewart, G. (2009). Reference governor for constrained piecewise affine systems. *Journal of Process Control*, 19(8), 1229–1237.

Folea, S., Muresan, C. I., Keyser, R. D., & Ionescu, C. M. (2016). Theoretical analysis and experimental validation of a simplified fractional order controller for a magnetic levitation system. *IEEE Transactions on Control Systems Technology*, 24(March (2)), 756–763.

Gaas, S., & Saaty, T. (1955). The computational algorithm for the parametric objective function. *Naval Research Logistics Quarterly*, 2, 39–45.

Geyer, T., Torrisi, F., & Morari, M. (2008). Optimal complexity reduction of polyhedral piecewise affine systems. *Automatica*, 44(July (7)), 1728–1740.

Gilbert, E. G., & Kolmanovsky, I. (1999). Fast reference governors for systems with state and control constraints and disturbance inputs. *International Journal of Robust Nonlinear Control*. http://dx.doi.org/10.1002/(SICI)1099-1239(19991230)9:15 < 1117::AID-RNC447 > 3.0.CO;2-I.

Glück, T., Kemmetmüller, W., Tump, C., & Kugi, A. (2011). A novel robust position estimator for self-sensing magnetic levitation systems based on least squares identification. *Control Engineering Practice*, 19(2), 146–157.

Gupta, A., Bhartiya, S., & Nataraj, P. (2011). A novel approach to multiparametric quadratic programming. *Automatica*, 47(9), 2112–2117.

Haisler, W. L., Timm, D. M., Gage, J. A., Tseng, H., Killian, T. C., & Souza, G. R. (2013). Three-dimensional cell culturing by magnetic levitation. *Nature Protocols*, 8, 1940–1949.

Herceg, M., Kvasnica, M., Jones, C., & Morari, M., 2013. Multi-parametric toolbox 3.0. In 2013 *European control conference* (pp. 502–510).

Kalúz, M., Klaučo, M., & Kvasnica, M. (2015). Real-time implementation of a reference governor on the Arduino microcontroller. In *Proceedings of the 20th international conference on process control* (pp. 350–356), Štrbské Pleso, Slovakia, June 9–12.

Klaučo, M. (2016). Modeling of the closed-loop system with a set of PID controllers, November. URL ⟨http://www.kirp.chtf.stuba.sk/publication_info.php?id_pub=1770⟩

Lee, H.-W., Kim, K.-C., & Lee, J. (2006). Review of maglev train technologies. *IEEE Transactions on Magnetics*, 42(July (7)), 1917–1925.

Lin, C. M., Lin, M. H., & Chen, C. W. (2011). Sopc-based adaptive pid control system design for magnetic levitation system. *IEEE Systems Journal*, 5(June (2)), 278–287.

Lin, C. M., Liu, Y. L., & Li, H. Y. (2014). Sopc-based function-link cerebellar model articulation control system design for magnetic ball levitation systems. *IEEE Transactions on Industrial Electronics*, 61(August (8)), 4265–4273.

Löfberg, J. (2004). YALMIP: A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD conference*, Taipei, Taiwan.

Maciejowski, J. M. (2002). *Predictive control with constraints* Essex, England: PEARSON Prentice-Hall.

Mayne, D., Rawlings, J., Rao, C., & Scokaert, P. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.

Pistikopoulos, E., Diangelakis, N., Oberdieck, R., Papathanasiou, M., Nascu, I., & Sun, M. (2015). PAROC: An integrated framework and software platform for the optimisation and advanced model-based control of process systems. *Chemical Engineering Science*, 136, 115–138.

Pistikopoulos, E. N., Dua, V., Bozinis, N. A., Bemporad, A., & Morari, M. (2002). On-line optimization via off-line parametric optimization tools. *Computers & Chemical Engineering*, 26(2), 175–185.

Schuhmann, T., Hofmann, W., & Werner, R. (2012). Improving operational performance of active magnetic bearings using Kalman filter and state feedback control. *IEEE Transactions on Industrial Electronics*, 59(February (2)), 821–829.

Tøndel, P., Johansen, T. A., & Bemporad, A. (2003). Evaluation of piecewise affine control via binary search tree. *Automatica*, 39(May (5)), 945–950.

Zhang, Y., Xian, B., & Ma, S. (2015). Continuous robust tracking control for magnetic levitation system with unidirectional input constraint. *IEEE Transactions on Industrial Electronics*, 62(September (9)), 5971–5980.

Applied Thermal Engineering 110 (2017) 1437–1447

Contents lists available at ScienceDirect

Applied Thermal Engineering

journal homepage: www.elsevier.com/locate/apthermeng

Research Paper

Control of a boiler-turbine unit using MPC-based reference governors

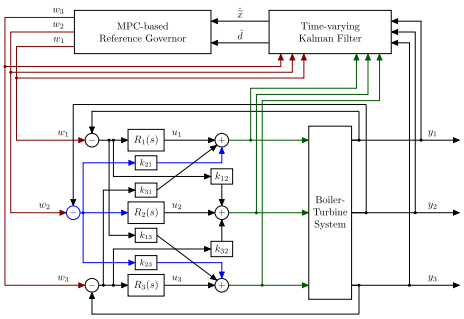
Martin Klaučo *, Michal Kvasnica

Slovak University of Technology in Bratislava, Radlinského 9, SK-812 37 Bratislava, Slovak Republic

HIGHLIGHTS

- Synthesis of an MPC-based reference governor.
- Modeling of a closed-loop with set of PI controllers.
- Control of a Boiler-Turbine Unit via MPC of the closed-loop.

GRAPHICAL ABSTRACT



ARTICLE INFO

Article history:
Received 9 December 2015
Revised 19 August 2016
Accepted 9 September 2016
Available online 10 September 2016

Keywords:
Boiler-turbine unit
Reference governor
Model predictive control
Optimization

ABSTRACT

An optimization-based scheme is proposed to improve safety and economical performance of a boiler-turbine system controlled by a set of interconnected PI controllers. The scheme, referred to as a reference governor, optimizes the references fed to the PI controllers in such a way that constraint satisfaction is enforced and tracking performance is improved. The proposed solution allows the plant operator to keep the existing PI control architecture and improve its performance by using an additional layer of control. The reference governor is based on optimizing the predicted future evolution of the closed-loop system, which consists of the plant and the inner PI controllers. Disturbance modeling is used to reject influence of the model-plant mismatch and to improve tracking performance. A case study is offered to demonstrate viability of the proposed approach with respect to energy savings and plant safety.

1. Introduction

The concept of optimization-based controllers has attracted the interest of process industries since early 1980s [1]. Such algorithms proved to be very effective in reducing the operating costs as well as increasing the safety of the entire plant operation [2]. Such a level of economic and safe operation is difficult to achieve using traditional control loops, which typically involve PI/PID controllers [3]. Therefore optimization-based control strategies, such as those based on Model Predictive Control (MPC) [4], are preferred in many areas, such as in petrochemical industries. Predictive control together with optimization plays important role also in power industries, where such approaches are used to increase power generation by tens of per cents [5]. Power generation and power optimization closely relates to smart energy systems and control. In such setups control of individual components may become challenging [6], hence methods like machine learning or big-data approaches are considered [7,8].

MPC offers several crucial advantages over conventional control loops based on PI/PID controllers. First, MPC provides a safe operation of the plant *by construction* since it explicitly accounts for constraints in the decision-making process. PI/PID approaches, on the other hand, have difficulties in handling of constraints. Their

* Corresponding author.
E-mail address: martin.klauco@stuba.sk (M. Klaučo).

http://dx.doi.org/10.1016/j.applthermaleng.2016.09.041
1359-4311/

influence is traditionally compensated by devising additional control logic represented by anti-windup schemes, or by detuning of the controllers. Both solutions, however, degrade the performance of the feedback loop. MPC strategies, on the other hand, improve the economical performance of the plant by optimizing a suitable performance criterion. Finally, MPC controllers have the natural ability to handle multivariable systems without the need to use decoupling schemes.

Despite the advantages reported above, some industries are still reluctant to replace their existing controllers by MPC-based control policies due to a mixture of factors. One of them is the desire to keep existing PI/PID controllers in place since they are easy to understand for process engineers and because a significant amount of effort was invested towards their design and tuning. In this paper we therefore show how to apply optimization-based control *on top* of existing PI/PID loops using the concept of *reference governors* [9–11]. Such governors act as a supervisory layer and modify the reference fed to the inner PI/PID controllers in such a way that even the simple inner controllers enforce constraint satisfaction. Moreover, by optimizing the reference signals, the overall performance of the plant is improved [12,13]. In fact, the concept of reference governors is not new. It can be traced back to early implementation of MPC algorithms, which were applied in exactly his manner [14,1]. The difference here, however, is that early implementation of MPC-based reference governors relied mostly on heuristic solutions to the control problems, which leads to suboptimal performance. In this work, following more recent concepts of [15–17], we formulate and solve the reference governor problem as an optimal control problem. This, combined with using off-the-shelve optimization solvers, yields better performance and improves process safety. The latter is due to the process constraints being explicitly accounted for in the optimization problem.

In this paper we show how to design a reference governor for a well-known boiler-turbine system, introduced in [18]. It represents a system where fossil fuel is burned to generate steam in a drum boiler. The steam is subsequently fed into the turbine. The system features three controlled outputs and three manipulated variables, which have to be operated subject to constraints on their respective amplitudes and their slew rates. Various strategies have been proposed in the literature to control such a plant, ranging from the application of fuzzy MPC [19,20], through data-driven approaches [21], dynamic matrix control [22], up to the application of hybrid MPC techniques [23,24]. Although all aforementioned approaches can substantially improve safety and profitability of the plant operation, they also assume that the existing control architecture (represented by the inner PI/PID controllers) is completely replaced by the new setup. As mentioned above, this is not always desired by plant operators.

In this paper we assume that the individual manipulated variables are controlled by a set of PI loops which, however, do not explicitly take constraints and performance objectives into account. Therefore, an optimization-based reference governor setup is proposed to improve upon these two factors. We show that the inner PI control loops can be modeled as a discrete-time linear time-invariant system. Then, we design a suitable state observer to estimate values of the states which cannot be directly measured, as well as to estimate unmeasured disturbances. Using the model and the estimates we then formulate the optimization problem which predicts the future evolution of the inner closed-loop system and optimizes references provided to the inner PI controllers. Offset-free tracking of output references is furthermore improved by using the concept of disturbance modeling [25]. By means of a case study we illustrate that the proposed concept has two main advantages. First, it allows to keep existing control infrastructure. Second, and more importantly, it enforces a safe and economic operation of the plant. The case study reported in this paper quantifies the improvement in safety and profitability and also compares it to the scenario where the plant is directly controlled by an MPC controller which bypasses existing PI-based control infrastructure.

2. Plant description and control objectives

We consider the well-known fossil-fueled boiler-turbine benchmark, presented in [18]. The system consists of a boiler where water is heated by burning fossil fuel and converted into steam. The steam is subsequently fed into a turbine which generates electricity. The system is controlled by three manipulated variables: the fuel valve, the feed-water valve, and the steam valve. The control variables are represented by the liquid level in the boiler, the steam pressure, and the generated power. A schematic representation of the plant is shown in Fig. 1.

The three manipulated variables of the boiler-turbine system are individually controlled by a set of three interconnected PI controllers as suggested by [26]. Their primary purpose is to stabilize the plant. However, they may exhibit a poor tracking performance due to presence of constraints. Specifically, the amplitude of each control action is constrained in the (normalized) interval $[0,1]$. Additionally, slew rate constraints have to be considered to guarantee a physically safe operation of the plant. Although anti-windup logic can be included into the feedback law to mitigate the influence of min/max constraints on the amplitude, dealing with slew rate constraints requires extensive controller tuning, which is a tedious procedure without any rigorous guarantees of being successful.

In this paper we propose to enforce constraint satisfaction and to improve tracking performance by devising a suitable reference governor. Its purpose is to shape the references provided to individual PI controllers such that even these simple controllers achieve constraints satisfaction and desirable performance. The schematic representation of the proposed strategy is shown in Fig. 2. Specifically, the reference governor replaces the user-specified reference r by a shaped reference signal w in such a way that the control actions u generated by the inner PI controllers respect constraints. Moreover, the reference governor improves tracking performance by taking into account predictions of the future evolution of the inner closed-loop system such that the measured plant's output y_{m} converges to the user-specified reference r without a steady-state offset.

The objective of this paper is to first demonstrate the design of a suitable reference governor. We show that it can be designed as an MPC-like optimization problem solution of which are the optimally shaped references w that enforce constraint satisfaction and optimize tracking performance. Subsequently, in the second part of the paper, the performance of the proposed strategy is compared to a pure PI-based control strategy as well as to full-fledged MPC setup by means of a case study.

3. Plant modeling and constraints

The mathematical model of the boiler-turbine plant was proposed in [18] as a set of three nonlinear differential equations of the form

$$\frac{\mathrm{d}p}{\mathrm{d}t} = -0.0018u_2p^{\frac{9}{8}} + 0.9u_1 - 0.15u_3, \tag{1a}$$

$$\frac{\mathrm{d}P_{\mathrm{N}}}{\mathrm{d}t} = (0.073u_2 - 0.016)p^{\frac{9}{8}} - 0.1P_{\mathrm{N}}, \tag{1b}$$

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = \frac{1}{85}(141u_3 - (1.1u_2 - 0.19)p). \tag{1c}$$

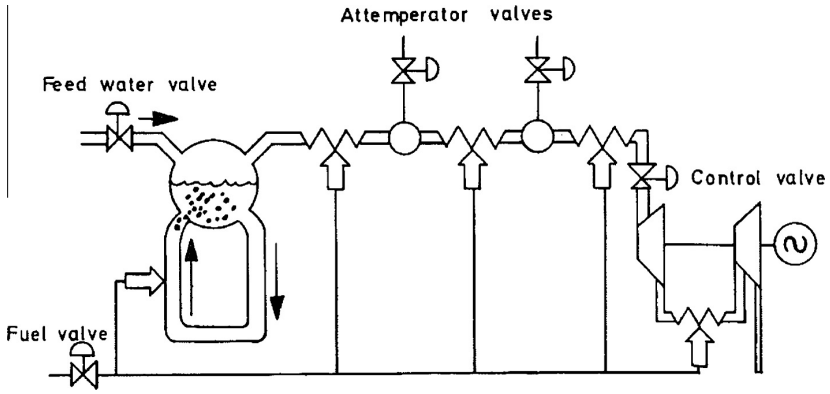


Fig. 1. The boiler-turbine plant, picture reproduced from [18].

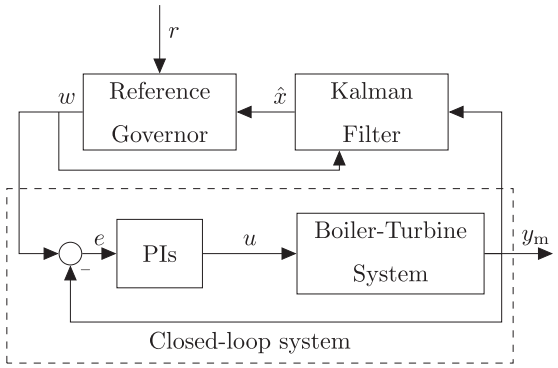


Fig. 2. Reference governor setup.

Here, p is the drum pressure in kg cm^{-2}, P_N denotes the nominal power generated by the turbine in MW, and ρ is the density of the liquid in the boiler in kg m^{-3}. These three variables represent the three states of the system. Moreover, the model features three control inputs: u_1 is the fuel flow valve, u_2 represents the steam valve, and u_3 denotes the feed-water valve. The liquid level h in the boiler, represented in m, is given as a nonlinear function of states and inputs in the form

$$h = 6.34 \times 10^{-3} p + 4.71 \times 10^{-3} \rho + 0.253 u_1 + 0.512 u_2 - 0.014 u_3. \quad (2)$$

Only h, p, and P_N can be directly measured. The density ρ is an internal state which can only be estimated if it is required by the control strategy.

The three control inputs are subject to min/max bounds on their amplitude with $0 \leqslant u_i \leqslant 1, i = 1, \ldots, 3$, where 0 represents the fully closed position and 1 stands for the fully open position. In addition, slew rate constraints must be respected as well. They are given by

$$\left|\frac{du_1}{dt}\right| \leqslant 0.007 \text{ s}^{-1}, \quad (3a)$$

$$-2 \text{ s}^{-1} \leqslant \frac{du_2}{dt} \leqslant 0.02 \text{ s}^{-1}, \quad (3b)$$

$$\left|\frac{du_3}{dt}\right| \leqslant 0.05 \text{ s}^{-1}. \quad (3c)$$

More details about this particular nonlinear model as well as operation of the boiler-turbine unit itself can be found in [26].

The reference governor design presented in this paper is based on a linearization of the nonlinear dynamics in (1), followed by conversion of the continuous-time model into the discrete-time domain. Specifically, let

$$x = [p \quad P_N \quad \rho]^T - x_L, \quad (4a)$$

$$u = [u_1 \quad u_2 \quad u_3]^T - u_L, \quad (4b)$$

$$y = [p \quad P_N \quad h]^T - y_L, \quad (4c)$$

denote, respectively, the vector of states, control actions, and measured outputs, expressed as deviations from respective linearization points x_L, u_L, and y_L. Then the linear time-invariant (LTI) approximation of the system in (1) in the discrete-time domain is given by

$$x(t + T_s) = Ax(t) + Bu(t), \quad (5a)$$

$$y(t) = Cx(t) + Du(t), \quad (5b)$$

where T_s is the sampling time. In this paper we assume $T_s = 2$ s.

The linearization point was selected as

$$x_L = [107.97 \quad 66.62 \quad 428.00]^T, \quad (6a)$$

$$u_L = [0.34 \quad 0.69 \quad 0.44]^T, \quad (6b)$$

$$y_L = [107.97 \quad 66.62 \quad 3.13]^T, \quad (6c)$$

and corresponds to the steady state where the turbine generates 66.62 MW of power with drum pressure of 107.97 kg cm^{-2}, and liquid density 428.00 kg m^{-3}.

Then, using the first-order Taylor approximation of the nonlinearities in (1) and by applying the forward Euler discretization, we arrive at the following matrices of the LTI model in (5):

$$A = \begin{bmatrix} 0.9950 & 0 & 0 \\ 0.1255 & 0.8187 & 0 \\ -0.0134 & 0 & 1 \end{bmatrix}, \quad (7a)$$

$$B = \begin{bmatrix} 1.7955 & -0.6963 & -0.2993 \\ 0.1168 & 25.6134 & -0.0195 \\ -0.0120 & -2.7733 & 3.3200 \end{bmatrix}, \quad (7b)$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.0063 & 0 & 0.0047 \end{bmatrix}, \quad (7c)$$

$$D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.2530 & 0.5120 & -0.0140 \end{bmatrix}. \quad (7d)$$

Worth noting is that the D matrix is non-zero, which is a consequence of direct feedthrough of control inputs in the output Eq. (2).

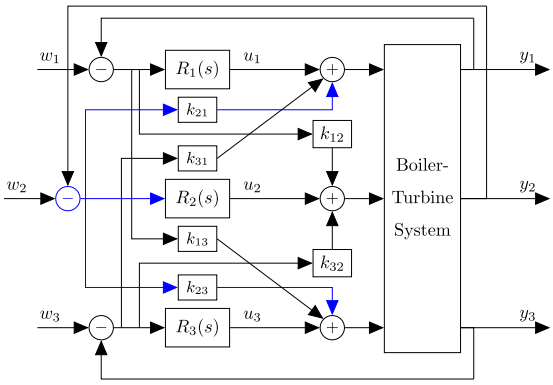


Fig. 3. Interconnected PI controllers as proposed by [26].

4. Modeling of the closed-loop system

We assume that the boiler-turbine system is controlled by three interconnected PI controllers as shown in Fig. 3. Their coefficients, as reported by [26], are as follows:

$$R_1 = \frac{11.119s + 0.003}{s}, \tag{8a}$$

$$R_2 = \frac{0.004s + 0.009}{s}, \tag{8b}$$

$$R_3 = \frac{1.163s + 0.019}{s}. \tag{8c}$$

Furthermore, local gains $k_{12} = 0.0292, k_{13} = 0.1344$, $k_{21} = 0.0468, k_{23} = 0.0875, k_{31} = 0.0842$ and $k_{32} = 0.0699$ are introduced to improve the control performance.

To deal with constraints [26], have proposed to include additional rate limiters and anti-windup logic into the feedback loop. In this paper this additional logic is not required since constraints satisfaction and performance criteria will be enforced by the reference governor. Its design will be based on an LTI representation of the closed-loop system shown in Fig. 2. To obtain the LTI form of the closed-loop system, we first derive the LTI representation of the inner PI-based control strategy shown in Fig. 3:

$$x_r(t + T_s) = A_r x_r(t) + B_r e(t), \tag{9a}$$

$$u(t) = C_r x_r(t) + D_r e(t). \tag{9b}$$

Here, x_r are the states of the three PI controllers, $e(t) = w(t) - y(t)$ is the vector of tracking errors (which are the inputs to the controllers), and $u(t)$ is the vector of control actions devised by the controllers as their respective outputs.

By using basic rules of algebra of transfer functions, and by applying discretization with $T_s = 2$ s we obtain the following representation of the matrices of the model in (9):

$$A_r = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{10a}$$

$$B_r = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \tag{10b}$$

$$C_r = \begin{bmatrix} 0.0033 & 0 & 0 \\ 0 & 0.0093 & 0 \\ 0 & 0 & 0.0186 \end{bmatrix}, \tag{10c}$$

$$D_r = \begin{bmatrix} 11.1185 & 0.0468 & 0.0842 \\ 0.0292 & 0.0040 & 0.0699 \\ 0.1344 & 0.0875 & 1.1631 \end{bmatrix}. \tag{10d}$$

Note that the D_r is a full matrix, which is a consequence of using local gains k_{ij} in the setup of Fig. 3.

Finally, the aggregated model of the inner closed-loop system in Fig. 2, which consists of the PI controllers and the plant, can be derived as follows. Consider the aggregated state vector

$$\widetilde{x} = \begin{bmatrix} x_r \\ x \end{bmatrix}, \tag{11}$$

which consists of the states of the PI controllers, i.e., x_r, and the states of the system in (5), denoted by x.

The closed-loop dynamics is then captured by

$$\widetilde{x}(t + T_s) = A_{CL}\widetilde{x}(t) + B_{CL}w(t), \tag{12a}$$

$$u(t) = C_{CL,u}\widetilde{x}(t) + D_{CL,u}w(t), \tag{12b}$$

$$y(t) = C_{CL,y}\widetilde{x}(t) + D_{CL,y}w(t). \tag{12c}$$

The inputs to this aggregated system are represented by w, the vector of reference to the PI controllers. The model features two output equations with (12b) representing the control actions generated by the inner PI controllers, and (12c) denoting the plant's outputs.

In order to obtain the system matrices in (12), first let us formulate an open loop state space model, where the input to the system is control error $e(t)$ and the outputs are control inputs to the plant $u(t)$ and plants' output $y(t)$. The open loop state space model is

$$\widetilde{x}(t + T_s) = A_{OL}\widetilde{x}(t) + B_{OL}e(t), \tag{13a}$$

$$u(t) = C_{OL,u}\widetilde{x}(t) + D_{OL,u}e(t), \tag{13b}$$

$$y(t) = C_{OL,y}\widetilde{x}(t) + D_{OL,y}e(t) \tag{13c}$$

where

$$A_{OL} = \begin{bmatrix} A_r & 0 \\ BC_r & A \end{bmatrix}, \tag{14a}$$

$$B_{OL} = \begin{bmatrix} B_r \\ BD_r \end{bmatrix}, \tag{14b}$$

$$C_{OL,u} = [C_r \quad 0], \tag{14c}$$

$$D_{OL,u} = D_r, \tag{14d}$$

$$C_{OL,y} = [DC_r \quad C], \tag{14e}$$

$$D_{OL,y} = DD_r. \tag{14f}$$

Second, combining the output Eq. (13c) and the expression of tracking error

$$e(t) = w(t) - y(t), \tag{15}$$

we arrive at

$$y(t) = (I + D_{OL,y})^{-1}C_{OL,y}\widetilde{x}(t) + (I + D_{OL,y})^{-1}D_{OL,y}w(t). \tag{16}$$

After inserting (15) and (16) into (13) we obtain

$$A_{CL} = A_{OL} - B_{OL}(I + D_{OL,y})^{-1}C_{OL,y}, \tag{17a}$$

$$B_{CL} = B_{OL} - B_{OL}(I + D_{OL,y})^{-1}C_{OL,y}, \tag{17b}$$

$$C_{CL,u} = C_{OL,u} - D_{OL,u}(I + D_{OL,y})^{-1}C_{OL,u}, \tag{17c}$$

$$D_{CL,u} = D_{OL,u} - D_{OL,u}(I + D_{OL,y})^{-1}C_{OL,u}, \tag{17d}$$

$$C_{CL,y} = C_{OL,y}(I + D_{OL,y})^{-1}, \tag{17e}$$

$$D_{CL,y} = D_{OL,y}(I + D_{OL,y})^{-1}. \tag{17f}$$

This state-space model of the closed-loop system will be used in the subsequent section to design an optimization-based reference governor.

5. Synthesis of an optimization-based reference governor

In this section the model of the closed-loop system in (12) with the matrices in (17) will be used to devise a reference governor

strategy which shapes the references $w(t)$ in such a way that the control commands of the inner PI controllers, i.e., $u(t)$ via (12b) satisfy all design constraints. Moreover, the plant's outputs $y(t)$ will be forced to converge to the user-specified reference $r(t)$ without an offset.

Since the governor will employ the state-space model, first we describe the design of a state estimator in Section 5.1. The estimator serves two purposes. First, it will estimate unmeasured components of the extended state vector (11), i.e., the density of the liquid in the drum ρ, and the states of the PI controllers x_r. Moreover, the estimator will also estimate unmeasured disturbances which will serve to reject the plant-model mismatch and thus allow for an offset-free tracking even when the plant operates away from the selected operating point, represented by (6). Then, in Section 5.2 we show how the reference governor problem is set up and solved.

5.1. State estimation and disturbance modeling

To reject the plant-model mismatch and to estimate states of the closed-loop system (12) we use the disturbance modeling approach of [25]. The theoretical principle of the disturbance modeling lies in estimating one unmeasured disturbance $d_i(t)$ for each plant output signal $y_i(t)$ which needs to track a user-defined reference signal $r_i(t)$. Then a time-varying Kalman filter is employed to estimate the state variables of (12), as well as the vector of unmeasured disturbances $d(t)$. In our case, $d \in \mathbb{R}^3$ since we want to control three outputs of the plant (the drum pressure, the nominal power, and the liquid level in the drum).

Consider the extended state-space model of the closed-loop system in (12), described by

$$\widetilde{x}_{k+1} = A_{\mathrm{CL}}\widetilde{x}_k + B_{\mathrm{CL}}w_k, \tag{18a}$$
$$u_k = C_{\mathrm{CL,u}}\widetilde{x}_k + D_{\mathrm{CL,u}}w_k, \tag{18b}$$
$$y_k = C_{\mathrm{CL,y}}\widetilde{x}_k + D_{\mathrm{CL,y}}w_k + Fd_k, \tag{18c}$$
$$d_{k+1} = d_k, \tag{18d}$$

where the unmeasured disturbances d enter into (18c) via a user-specified matrix F. Moreover, the disturbances are assumed to have constant dynamics, cf. (18d), see [27]. The subindex k denotes the predictions of the corresponding variable in the estimation problem. Define the estimated extended state vector as

$$\hat{x}_{\mathrm{e}} = \begin{bmatrix} \hat{\widetilde{x}} \\ \hat{d} \end{bmatrix}, \tag{19}$$

where $\hat{\widetilde{x}}$ is the estimate of the state vector of the closed loop system in (12) (cf. (11)), and $\hat{d}$ is the estimate of the unmeasured disturbances.

The time-varying Kalman filter procedure consists of two phases. The first phase is the prediction phase which is followed by the update phase. In both phases the index $k|k$ represents current estimate of the individual variables. The prediction phase consist of two equations, namely

$$\hat{x}_{\mathrm{e},k|k-1} = A_{\mathrm{e}}\hat{x}_{\mathrm{e},k-1|k-1} + B_{\mathrm{e}}w_k, \tag{20a}$$
$$P_{k|k-1} = A_{\mathrm{e}}P_{k-1|k-1}A_{\mathrm{e}}^{\mathsf{T}} + Q_{\mathrm{e}} \tag{20b}$$

where $\hat{x}_{\mathrm{e},k|k-1}$ is the predicted state estimate based on the previous time instant, and $P_{k|k-1}$ is the predicted value of the covariance matrix. The matrices $A_{\mathrm{e}}, B_{\mathrm{e}}, C_{\mathrm{e}}, D_{\mathrm{e}}$ are given by

$$A_{\mathrm{e}} = \begin{bmatrix} A_{\mathrm{CL}} & 0 \\ 0 & I \end{bmatrix}, \quad B_{\mathrm{e}} = \begin{bmatrix} B_{\mathrm{CL}} \\ 0 \end{bmatrix}, \tag{21a}$$
$$C_{\mathrm{e}} = \begin{bmatrix} C_{\mathrm{CL,u}} & 0 \\ C_{\mathrm{CL,y}} & F \end{bmatrix}, \quad D_{\mathrm{e}} = \begin{bmatrix} D_{\mathrm{CL,u}} \\ D_{\mathrm{CL,y}} \end{bmatrix}, \tag{21b}$$

In theory, the matrix Q_{e} used in (20) should be chosen with the respect to the stochastic properties of the state noise signal. However, in this work we only consider deterministic simulations, therefore this matrix is chosen as a tuning parameter of the time-varying Kalman Filter.

Remark 1. In our setup, all states of the closed-loop system along with the disturbances are estimated, cf. (19). Note that $\widetilde{x}$ in (19) includes also the internal states of the inner PI controllers x_{r} via (11). If complexity of the time-varying Kalman is of concern, these known states can be removed from the estimator and replaced by actual measurements. However, as pointed out by several authors (see, e.g., [27–30]), estimation of all states (including the known ones) can significantly mitigate the steady-state offset and improve the control performance.

The consecutive step in the estimation algorithm is the update phase, represented by

$$\epsilon_k = (y_{\mathrm{m},k} - y_{\mathrm{L}}) - (C_{\mathrm{e}}\hat{x}_{\mathrm{e},k|k-1} + D_{\mathrm{e}}w_k), \tag{22a}$$
$$S_k = C_{\mathrm{e}}P_{k|k-1}C_{\mathrm{e}}^{\mathsf{T}} + R_{\mathrm{e}}, \tag{22b}$$
$$L_k = P_{k|k-1}C_{\mathrm{e}}^{\mathsf{T}}S_k^{-1}, \tag{22c}$$
$$\hat{x}_{\mathrm{e},k|k} = \hat{x}_{\mathrm{e},k|k-1} + L_k\epsilon_k, \tag{22d}$$
$$P_{k|k} = (I - L_kC_{\mathrm{e}})P_{k|k-1}. \tag{22e}$$

Here, the estimation error ϵ_k is calculated based on the plant measurements $y_{\mathrm{m},k}$ per (22a). Since the estimator runs in deviation variables, the linearisation point y_{L} must be substracted from the measurement. The time-varying estimator gain L_k is then calculated by (22b) and (22c). This gain is subsequently used to obtain the current estimate of the state variables $\hat{x}_{\mathrm{e},k|k}$ via (22d). At the end of the update phase, the covariance matrix P is updated. The tuning matrix R_{e} should be chosen with respect to the stochastic properties of the output signals.

The estimates of $\widetilde{x}$ and d can then be extracted from $\hat{x}_{\mathrm{e}}$ by

$$\hat{\widetilde{x}}_k = M_{\mathrm{x}}\hat{x}_{\mathrm{e},k|k} \quad \hat{d}_k = M_{\mathrm{d}}\hat{x}_{\mathrm{e},k|k}, \tag{23}$$

where

$$M_{\mathrm{x}} = [\,I_{n_{\mathrm{x}}} \quad 0\,], \quad M_{\mathrm{d}} = [\,0 \quad I_{n_{\mathrm{d}}}\,]. \tag{24}$$

Here, n_{x} is the number of states of the closed-loop system in (12), n_{d} is the number of disturbances, and I is the identity matrix of corresponding dimension.

5.2. Reference governor design

In this section we show how to design a reference governor which enforces that the inner closed-loop system satisfies constraints and enforces sufficient control performance. The design is based on the principle of model predictive control where the predicted future evolution of the system is optimized as to achieve desired control objectives.

The prediction problem over a finite prediction horizon N is given by

$$\min_{w_0,\ldots,w_{N-1}} \quad \sum_{k=0}^{N-1} \|y_k - r\|_{Q_{\mathrm{y}}}^2 + \|\Delta w_k\|_{Q_{\mathrm{w}}}^2 + \|\Delta u_k\|_{Q_{\mathrm{u}}}^2 \tag{25a}$$
$$\text{s.t.} \quad \widetilde{x}_{k+1} = A_{\mathrm{CL}}\widetilde{x}_k + B_{\mathrm{CL}}w_k, \tag{25b}$$
$$u_k = C_{\mathrm{CL,u}}\widetilde{x}_k + D_{\mathrm{CL,u}}w_k, \tag{25c}$$
$$y_k = C_{\mathrm{CL,y}}\widetilde{x}_k + D_{\mathrm{CL,y}}w_k + Fd_0, \tag{25d}$$
$$y_{\min} \leqslant y_k \leqslant y_{\max}, \tag{25e}$$
$$u_{\min} \leqslant u_k \leqslant u_{\max}, \tag{25f}$$
$$\Delta u_{\min} \leqslant \Delta u_k \leqslant \Delta u_{\max}. \tag{25g}$$

The problem employs predictions of the states $\tilde{x}_k$, inputs u_k, and outputs y_k of the closed-loop system in (12), embedded via (25b)–(25d) for $k = 0, \ldots, N-1$. Moreover, the output Eq. (25d) uses a the disturbance d_0 which is constant over the whole prediction horizon.

The cost function in (25a) penalizes weighted squared 2-norms of respective quantities with $\|z\|_M^2 = z^\intercal M z$. The first term of the cost function minimizes the tracking error and forces the plant's outputs (the pressure, the power, and the liquid level) to track user-specified vector of references r. The second term penalizes fluctuations of the optimized references w with $\Delta w_k = w_k - w_{k-1}$, while the third terms minimizes the fluctuations of the control actions of the inner PI controllers with $\Delta u_k = u_k - u_{k-1}$. The cost function employs symmetric positive-definite tuning matrices Q_y, Q_w, and Q_u of suitable dimensions.

Future shaped references over the whole prediction horizon, i.e., $w_0, \ldots, w_{N-1}$, are optimized by (25) such that they enforce that the inner PI controllers' actions u_k respect constraints on their amplitudes and slew rates via (25f) and (25g), respectively. Constraints on the plant's outputs are also enforced via (25e). The optimization problem (25) is a quadratic program (QP) in decision variables $w_0, \ldots, w_{N-1}$ since the objective function (25a) is quadratic and all constraints in (25b)–(25g) are linear in the decision variables. The problem is initialized by:

- $\tilde{x}_0 = \tilde{x}(t)$ as the measurement (or estimate) of the states of the closed-loop system in (12) at the current time instant t;
- $d_0 = d(t)$ as the current estimate of unmeasured disturbances;
- $r = r(t)$ as the user-specified references for the controller outputs;
- $w_{-1} = w(t - T_s)$ as the value of the shaped reference at the previous sampling instant;
- $u_{-1} = u(t - T_s)$ as the value of the control actions of the inner PI controllers at the previous sampling instant.

Therefore the vector of initial conditions for (25) is

$$\theta = \left[\tilde{x}(t)^\intercal \quad d(t)^\intercal \quad r(t)^\intercal \quad w(t-T_s)^\intercal \quad u(t-T_s)^\intercal\right]^\intercal. \tag{26}$$

By solving (25) as a QP for a given vector of initial conditions θ we obtain the open-loop sequence of optimal shaped references $w_0^\star, \ldots, w_{N-1}^\star$ that enforce constraint satisfaction and provide that the controlled outputs track user-specified references.

5.3. Implementation

Notice that the optimization-based scheme for obtaining optimally shaped references w for the inner PI controllers is formulated using a finite prediction horizon N. Its recursive implementation in the closed-loop manner is performed using the principles of receding horizon control (RHC) [31]. Here, the optimization problem (25) is solved at each sampling instant for the initial conditions available at time step t, which yields the optimal open-loop sequence $w_0^\star, \ldots, w_{N-1}^\star$. Subsequently, only the first element of such a sequence, i.e., $w_0^\star$ is extracted and implemented to the plant for the duration of the sampling instant, given by T_s. The whole procedure is then repeated at any subsequent sampling instant using the measurements or estimates available at that time.

Formally, the RHC implementation of the proposed reference governor executes following steps at each sampling instant t:

1. Measure the current system output $y_m(t)$, which consists of the drum pressure, generated power, and the liquid level in the drum.
2. Estimate the current state $\hat{x}_e(t)$ using the Kalman filter. Set $\hat{\tilde{x}}(t) = M_x \hat{x}_e$ and $\hat{d}(t) = M_d \hat{x}_e$, cf. (23).
3. Solve (25) for $\tilde{x}_0 = \hat{\tilde{x}}(t), d_0 = \hat{d}(t), r = r(t), w_{-1} = w(t - T_s)$, $u_{-1} = u(t - T_s)$ and obtain the optimal open-loop sequence of shaped references $w_0^\star, \ldots, w_{N-1}^\star$.
4. Set $w^\star(t) = w_0^\star$ and apply it as a shaped reference to the inner PI controllers.

Such a recursive solution to (25) with the states and disturbances estimated using the Kalman filter will enforce satisfaction of constraints in (25f), (25g) and will steer system outputs towards prescribed references. Since a new instance of the QP is solved at each sampling instant, the reference signal $r(t)$ can be time-varying.

6. Case study

In this section we demonstrate the performance and viability of the proposed reference governor setup in a simulation study involving the nonlinear model of the boiler-turbine system in (1). Three cases are considered. In the first case the boiler is controlled solely by the interconnected PI controllers as shown in Fig. 3. In the second case the references for the inner PI controllers are shaped in an optimal fashion using the reference governor described in Section 5. We refer to this scenario as the RG-MPC setup. Finally, the last case considers that the plant is directly controlled by an MPC regulator, bypassing the inner PI loops. This scenario will be referred to as Direct-MPC and is shown in Fig. 4. The Direct-MPC case assumes that the optimal control actions for the boiler-turbine system are devised by solving the following MPC problem at each sampling instant:

$$\min_{u_0,\ldots,u_{N-1}} \quad \sum_{k=0}^{N-1} \|y_k - r\|_{W_y}^2 + \|\Delta u_k\|_{W_u}^2 \tag{27a}$$

$$\text{s.t.} \quad x_{k+1} = Ax_k + Bu_k, \tag{27b}$$

$$y_k = Cx_k + Du_k + Fd_0, \tag{27c}$$

$$y_{\min} \leqslant y_k \leqslant y_{\max}, \tag{27d}$$

$$u_{\min} \leqslant u_k \leqslant u_{\max}, \tag{27e}$$

$$\Delta u_{\min} \leqslant \Delta u_k \leqslant \Delta u_{\max}. \tag{27f}$$

Problem (27) is similar to (25), but we directly optimize the control inputs u of (5) instead of the shaped reference w. Moreover, the predictions are based on the model of the nominal system in (5). The MPC problem (27) can be implemented in a receding horizon fashion similarly as was described in Section 5.3. Its initial conditions, i.e., x_0 and d_0, are obtained by estimation after a suitable modification of the Kalman filtering approach by only considering the estimation of the states of the nominal system in (5) with the unmeasured disturbances included into the output equation.

In all three scenarios, the sampling time was set to $T_s = 2$ s and problems (25) and (27) were formulated for the prediction horizon

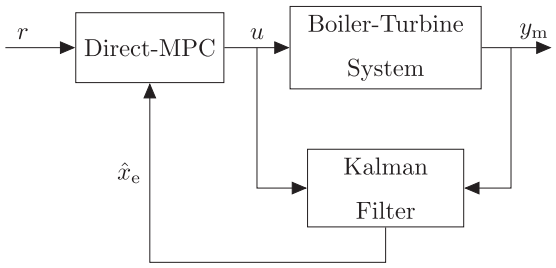


Fig. 4. Direct-MPC setup.

$N = 30$. The following tuning of penalty matrices used in (25a) was used:

$$Q_y = \mathrm{diag}([10^1\ 10^{-2}\ 10^1]),\ Q_w = I_{3\times3} \times 10^{-4},\ Q_u = I_{3\times3} \times 10^{-3}. \quad (28)$$

The associated Kalman filter was tuned with $P_0 = I_{9\times9}, Q_e = I_{9\times9} \times 10^{-4}$, and $R_e = I_{6\times6} \times 10^{-2}$.

In case of the direct MPC setup, represented by (27), we have used $W_y = \mathrm{diag}([10^2\ 10^{-1}\ 10^2])$ and $W_u = I_{3\times3} \times 10^{-3}$. The covariance matrix in time-varying Kalman filter was a unity matrix. Moreover,

$$Q_e = \mathrm{diag}([10^1\ 10^2\ 10^1\ 10^{-1}\ 10^1\ 10^{-1}])$$

and $R_e = \mathrm{diag}([10^1\ 10^1\ 10^{-5}])$ were used in the estimator.

Both optimization-based strategies were required to enforce following constraints:

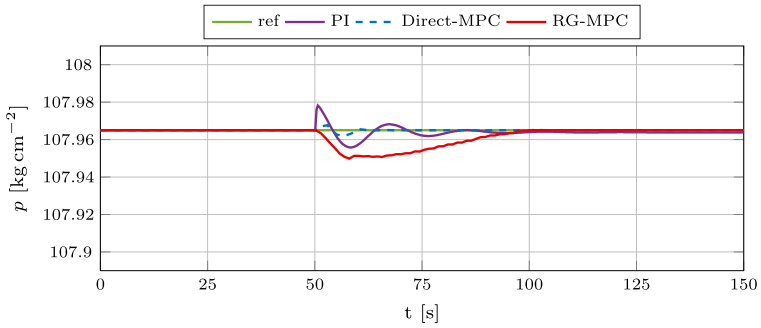
$$u_{\min} = [0\ 0\ 0]^\intercal - u_L, \quad u_{\max} = [1\ 1\ 1]^\intercal - u_L, \quad (29)$$

along with constraints on the slew rate of control inputs:

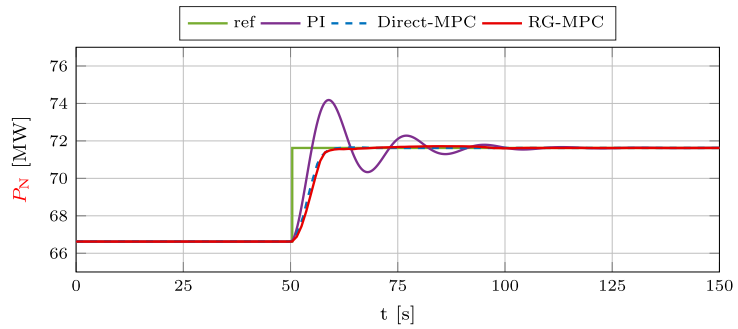
$$\Delta u_{\min} = [-0.014\ -4\ -0.1]^\intercal\ \mathrm{s}^{-1}, \quad (30a)$$

$$\Delta u_{\max} = [0.014\ 0.04\ 0.1]^\intercal\ \mathrm{s}^{-1}. \quad (30b)$$

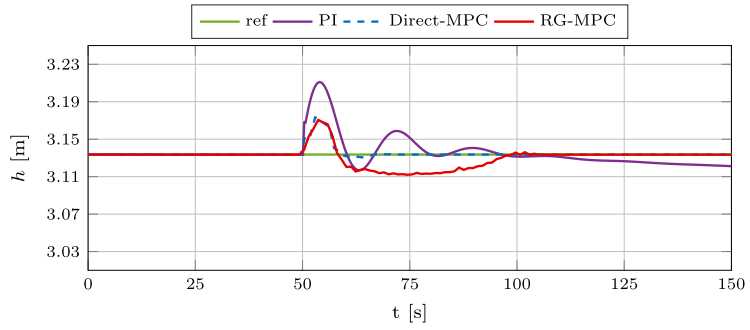
The first simulation scenario involves a step change in the requested generated power. Specifically, a +5 MW step change w. r.t. to the steady state $P = 66.62$ MW is performed at time 50 s. The objective is to keep the other two controlled outputs, i.e., the pressure p and the liquid level in the drum h at their



(a) Drum pressure



(b) Power generated by turbine



(c) Liquid level in boiler

Fig. 5. Measurements profiles. The green color depicts the reference signal. The pink color show the performance of set of PI controllers, the red lines depicts the performance using RG-MPC strategy, and the blue dashed is used for Direct-MPC setup. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

steady-state values. In all simulations the respective QP optimization problems were formulated using YALMIP [32] and solved using the Gurobi [33] solver.

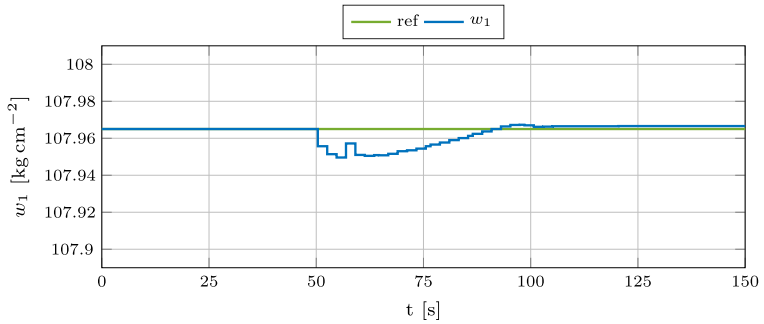
Closed-loop simulation profiles of the controlled outputs of the boiler-turbine system under the three scenarios are shown in Fig. 5. As can be observed, the power output signal exhibits noticeable oscillations when the system is controlled purely by the PI controllers. On the other hand, once their respective references are shaped via the reference governor, a much smoother tracking of the power reference is achieved. Worth noting is that by employing the disturbance modeling principle, perfect tracking of the references is achieved despite the controlled system (represented by the nonlinear plant in (1)) being different from the prediction model (represented by the LTI system (5)).

The corresponding profiles of the shaped references, generated by solving the QP (25) at each sampling instant, are shown in Fig. 6. Moreover, the performance of the proposed reference governor setup is almost identical to that one of the direct MPC. A similar conclusion can be drawn from the profiles of the liquid level in the boiler. The control actions of all three control strategies are shown in Fig. 7. As can be seen, the PI controllers are less aggressive such that they avoid hitting the constraints. The RG-MPC and Direct-MPC strategies, on the other hand, are explicitly aware of the constraints.

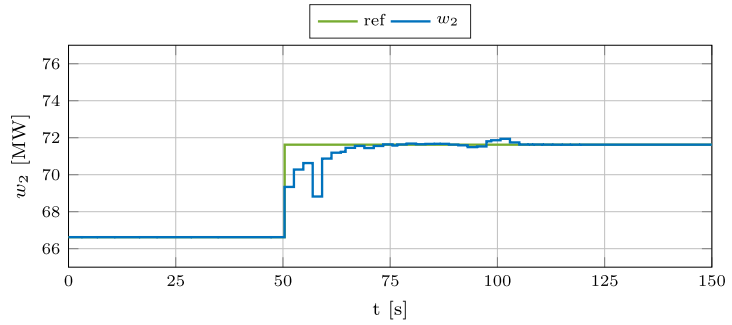
To quantify the performance of the three discussed control strategies, we have evaluated two criteria. The first one, represented by

$$E_{\mathrm{u}} = \frac{1}{N_{\mathrm{sim}}} \sum_{k=1}^{N_{\mathrm{sim}}} \|u_j(k)\|_1, \tag{31}$$

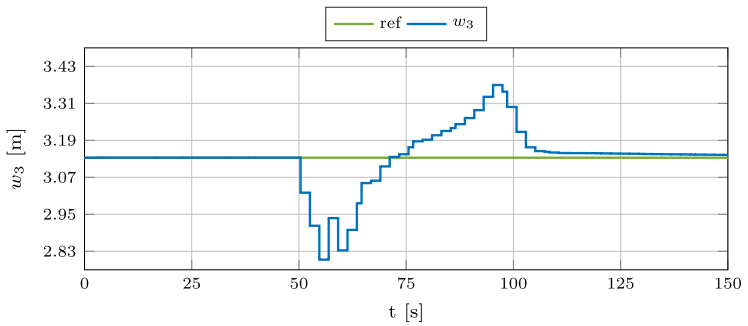
quantifies the amount of energy consumed by the individual control strategy. Here, N_{sim} is the number of simulation steps, $u(k)$ denotes



(a) Shaped reference for the drum pressure



(b) Shaped reference for the power setpoint



(c) Shaped reference for the liquid level in boiler

Fig. 6. Shaped references. The green color depicts the reference signal. The blue color show the shaped reference. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

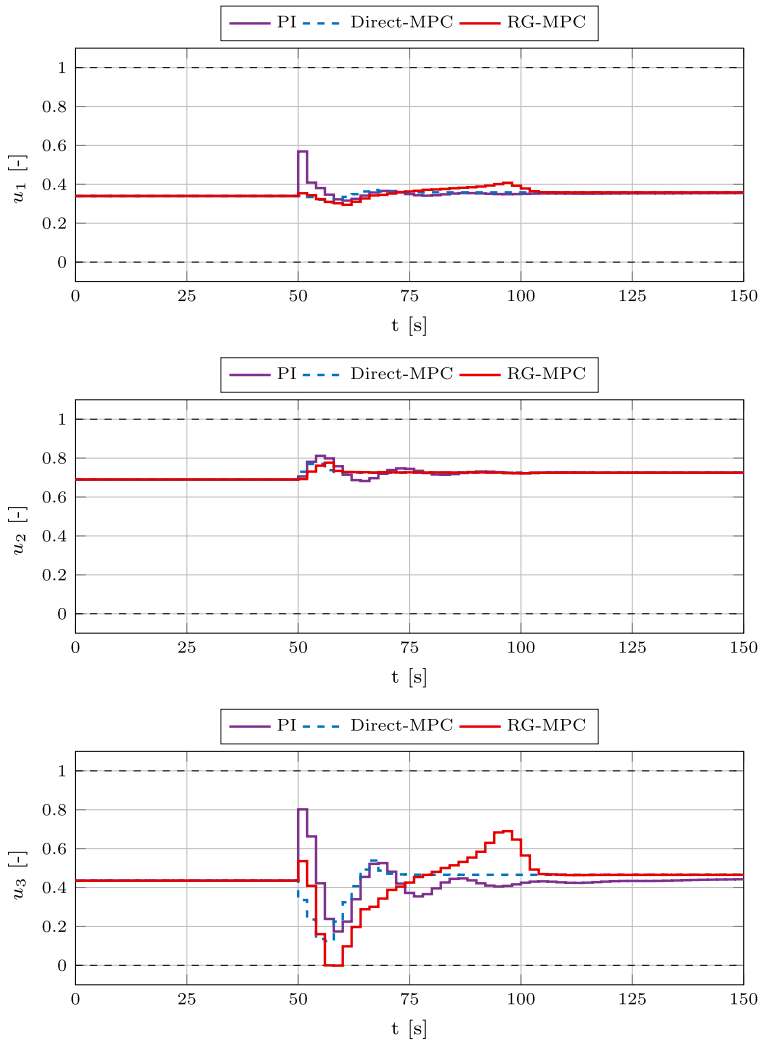


Fig. 7. Control actions profiles.

the control action at the k-th simulation step, and $\|u(k)\|_1 = \sum_{j=1}^{3}|u_j(k)|$ is the 1-norm of $u(k)$. Similarly,

$$E_y = \frac{1}{\frac{1}{N_{sim}}\sum_{k=1}^{N_{sim}}\|y(k) - r(k)\|_1} \tag{32}$$

quantifies the tracking performance.

Values of these performance criteria for each of the three considered control strategies are compared graphically in Fig. 8. Note that the values are normalized such that the PI strategy has $E_u = 1$ and $E_y = 1$.

As expected, the pure PI-based strategy exhibits the worst performance both with respect to energy consumption as well as in terms of tracking performance. This is a consequence of the conservative tuning of such PI controllers with the objective to avoid hitting constraints. The best performance is achieved by the Direct-MPC setup, since it bypasses the internal dynamics of the PI controllers. The proposed reference governor-based scheme allows the PI controller to stay as a part of the closed-loop system, but significantly improves their performance by a suitable choice of the shaped references. Specifically, compared to the pure PI strategy, RG-MPC reduces the energy consumption by 27% and improves the tracking performance by 48%.

To demonstrate the effect of tuning parameters used in (27a), besides the baseline setup of (28) we have also considered two different selection of the weighting matrices:

$$Q_y = \mathrm{diag}([10^2\ 10^{-1}\ 10^2]),\ Q_w = I_{3\times3}\times10^{-4},\ Q_u = I_{3\times3}\times10^{-5}, \tag{33a}$$

$$Q_y = \mathrm{diag}([5\times10^1\ 10^{-1}\ 10^1]),\ Q_w = I_{3\times3}\times10^{-5},\ Q_u = I_{3\times3}\times10^{-5}. \tag{33b}$$

The first tuning in (33a) puts a higher emphasis on tracking of references since the Q_y matrix is ten times as large as in (28). Moreover, Q_w and Q_u were decreased by a factor of 10. The second choice in (33b) allows the shaped reference to follow the user-supplied reference less tightly by further decreasing Q_w. As a consequence, the RG-MPC has more "freedom" in the choice of the shaped reference. In all cases, prediction horizon $N = 30$ was used as was the case in the baseline scenario.

The aggregated quality criteria (31) and (32) evaluated for these re-tuned RG-MPC strategies are depicted visually in Fig. 8. Moreover, concrete values of the quality criteria are reported in Table 1. Besides the aggregated quality criteria E_u and E_y computed

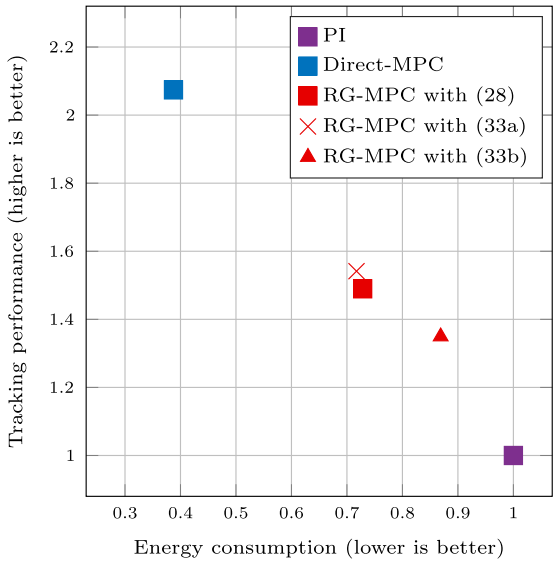


Fig. 8. Control strategies comparison.

Table 1
Quality criteria for various control strategies. Lower values of E_u and higher values of E_y are better. The values are normalized with respect to the PI strategy.

Strategy	E_u	E_y	$E_{y,p}$	E_{y,P_N}	$E_{y,h}$
PI	1.0000	1.0000	1.0000	1.0000	1.0000
Direct-MPC	0.3874	2.0741	7.5113	2.0341	5.7237
RG-MPC with (28)	0.7284	1.4897	0.7565	1.9189	0.1605
RG-MPC with (33a)	0.7173	1.5415	0.7652	2.0095	0.1601
RG-MPC with (33b)	0.8689	1.3494	0.7868	1.8572	0.1205

per (31) and (32), respectively, the table also reports tracking performance of the three controlled outputs. Specifically, $E_{y,p}$ related to the quality with which the drum pressure reference is tracked. Similarly, E_{y,P_N} is the quality of the tracking for the nominal power and $E_{y,h}$ relates to the quality of tracking of the liquid level reference.

Two main conclusions can be drawn from the results in Table 1. First, the best overall performance is achieved by using the tuning in (33a) which, however, is only marginally better than the baseline tuning of (28). The third alternative per (33b) is slightly worse with respect to output tracking and significantly worse w.r.t. energy consumption. The second conclusion is that all RG-MPC tunings provide a very good tracking quality of the nominal power P_N, even compared to the Direct-MPC setup, cf. the penultimate column of Table 1. Specifically, with the tuning in (33a), the tracking quality criterion E_{y,P_N} is only by 1.2% worse compared to Direct-MPC. Even with the tuning in (33b), which yields the worst tracking of the power reference, the deterioration of performance is only by 8.7%. On the other hand, by using only the PI controllers, the power tracking performance drops by 50.8% compared to Direct-MPC. Since the nominal power is considered the most important quality in practice, this demonstrates that RG-MPC can indeed significantly improve control quality even when inner PI controllers are included in the loop and can even achieve comparable results to Direct-MPC setups.

7. Conclusions

In this paper we have shown how to improve safety and performance of conventional control strategies by a suitable modification of their references. This was achieved by predicting the future evolution of the closed-loop system composed of the controlled plant and a set of interconnected PI controllers. By optimizing over the future predictions we have obtained shaped references which, when fed back to the inner controllers, lead to constraints satisfaction and improved tracking performance. The shaped references were computed by solving a convex quadratic programming problem at each sampling instant. The unmeasured states, as well as disturbances capturing the plant-model mismatch, were estimated using a time-varying Kalman filter. The case study has demonstrated that process safety and performance can indeed be significantly improved. The results of the proposed reference governor setup were also compared to a scenario where the inner controllers are bypassed and the plant is directly controlled by a model predictive control strategy.

By applying optimization on top of existing PI controllers, reference governors can be viewed at as “trojan horses” for application of MPC-based strategies to plants where complete revamp of the control architecture is not desired. This reduces the cost of implementation and, most importantly, allows human operators to stay comfortable with existing control architecture.

We see two main areas in which the RG-MPC concept can be further improved. The first one includes incorporation of a more detailed prediction model, preferably a nonlinear one. This, on the one hand, would result in further increase of control quality. On the other hand, however, computational obstacles associated with solving nonlinear optimization problems would need to be addressed. The second direction is to assume more complex inner controllers. For instance, one can assume that the coefficients of the PI controllers change based on a look-up table. This would result in a switching in the controller tuning. Such a behavior can be efficiently tackled in the context of hybrid systems [34] at the expense of arriving at a more complex optimization problem.

Acknowledgments

Authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under grant 1/0403/15 and the contribution of the Slovak Research and Development Agency under the project APVV-15-0007.

References

[1] C.R. Cutler, B.L. Ramaker, Dynamic matrix control – a computer control algorithm, Joint Automatic Control Conference, San Francisco, CA, USA, vol. 1, 1980.
[2] S.J. Qin, T.A. Badgewell, A survey of industrial model predictive control technology, Control Eng. Pract. 11 (2003) 733–764.
[3] K. Åström, T. Hägglund, Advanced PID Control, ISA – The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709, 2006.
[4] J.M. Maciejowski, Predictive Control with Constraints, Prentice-Hall, 2001.
[5] Yufeng Wang, Yanqing Niu, Xiaolu Zhang, Zhizhou Wang, Shuai Wang, Shien Hui, Optimization and energy integration of heat recovery and power generation system, Appl. Therm. Eng. 107 (2016) 294–300.
[6] Kaile Zhou, Shanlin Yang, Zhiqiang Chen, Shuai Ding, Optimal load distribution model of microgrid in the smart grid environment, Renew. Sustain. Energy Rev. 35 (2014) 304–310.
[7] Kaile Zhou, Chao Fu, Shanlin Yang, Big data driven smart energy management: from big data to big insights, Renew. Sustain. Energy Rev. 56 (2016) 215–225.
[8] Kai le Zhou, Shan lin Yang, Chao Shen, A review of electric load classification in smart grid environment, Renew. Sustain. Energy Rev. 24 (2013) 103–110.
[9] A. Bemporad, E. Mosca, Constraint fulfilment in feedback control via predictive reference management, in: Proceedings of the Third IEEE Conference on Control Applications, 1994, pp. 1909–1914.
[10] E. Gilbert, I. Kolmanovsky, K. Tan, Discrete-time reference governors and the nonlinear control of systems with state and control constraints, Int. J. Robust Nonlinear Control 5 (5) (1995) 487–504.
[11] E. Mosca, Nonlinear predictive command governors for constrained tracking, in: Colloquium on Automatic Control, 1996, pp. 55–76.
[12] A. Bemporad, Reference governor for constrained nonlinear systems, IEEE Trans. Autom. Control 43 (3) (1998) 415–419.

[13] E.G. Gilbert, I. Kolmanovsky, Nonlinear tracking control in the presence of state and control constraints: a generalized reference governor, Automatica 38 (12) (2002) 2063–2073.
[14] J. Richalet, A. Rault, J.L. Testud, J. Papon, Model predictive heuristic control, Automatica 14 (5) (1978) 413–428.
[15] F. Borrelli, P. Falcone, J. Pekar, G. Stewart, Reference governor for constrained piecewise affine systems, J. Process Control 19 (8) (2009) 1229–1237.
[16] E.G. Gilbert, I. Kolmanovsky, Fast reference governors for systems with state and control constraints and disturbance inputs, Int. J. Robust Nonlinear Control (1999).
[17] A. Bemporad, A. Casavola, E. Mosca, A predictive reference governor for constrained control systems, Comput. Ind. 36 (12) (1998) 55–64.
[18] K.J. Åström, K. Eklund, A simplified non-linear model of a drum boiler-turbine unit, Int. J. Control 16 (1) (1972) 145–169.
[19] X. Liu, X. Kong, Nonlinear fuzzy model predictive iterative learning control for drum-type boilerturbine system, J. Process Control 23 (8) (2013) 1023–1040.
[20] Y. Li, J. Shen, K. Lee, X. Liu, Offset-free fuzzy model predictive control of a boilerturbine system based on genetic algorithm, Simul. Model. Pract. Theory 26 (2012) 77–95.
[21] X. Wu, J. Shen, Y. Li, K. Lee, Data-driven modeling and predictive control for boilerturbine unit using fuzzy clustering and subspace methods, {ISA} Trans. 53 (3) (2014) 699–708.
[22] U. Moon, K.Y. Lee, Step-response model development for dynamic matrix control of a drum-type boiler-turbine system, IEEE Trans. Energy Convers. 24 (2) (2009) 423–430.
[23] M. Sarailoo, B. Rezaie, Z. Rahmani, MLD model of boiler-turbine system based on PWA linearization approach, Int. J. Control Sci. Eng. 2 (2012) 88–92.
[24] M. Keshavarz, M. Barkhordari Yazdi, M.R. Jahed-Motlagh, Piecewise affine modeling and control of a boilerturbine unit, Appl. Therm. Eng. 30 (89) (2010) 781–791.
[25] G. Pannocchia, J.B. Rawlings, Disturbance models for offset-free model-predictive control, AIChE J. 49 (2) (2003).
[26] R. Dimeo, K.Y. Lee, Boiler-turbine control system design using a genetic algorithm, IEEE Trans. Energy Convers. 10 (4) (1995) 752–759.
[27] K.R. Muske, T.A. Badgwell, Disturbance modeling for offset-free linear model predictive control, J. Process Control 12 (2002) 617–632.
[28] G. Pannocchia, E.C. Kerrigan, Offset-free Receding Horizon Control of Constrained Linear Systems subject to Time-varying Setpoints and Persistent Unmeasured Disturbances. Technical Report CUED/F-INFENG/TR.468, Department of Engineering, University of Cambridge, Cambridge, UK, 2004.
[29] G. Pannocchia, J.B. Rawlings, Disturbance models for offset-free model-predictive control, AIChE J. 49 (2) (2003) 426–437.
[30] G. Pannocchia, E.C. Kerrigan, Offset-free control of constrained linear discrete-time systems subject to persistent unmeasured disturbances, in: 42nd Conference on Decision and Control, Maui, Hawaii, USA, December, 2003.
[31] D.Q. Mayne, J.B. Rawlings, C.V. Rao, P.O.M. Scokaert, Constrained model predictive control: stability and optimality, Automatica 36 (6) (2000) 789–814.
[32] J. Löfberg, YALMIP: A Toolbox for Modeling and Optimization in MATLAB, in: Proc. of the CACSD Conference, Taipei, Taiwan, 2004, Available from http://users.isy.liu.se/johanl/yalmip/.
[33] Inc. Gurobi Optimization, Gurobi Optimizer Reference Manual, 2013.
[34] A. Bemporad, M. Morari, Control of systems integrating logic, dynamics, and constraints, Automatica 35 (3) (1999) 407–427.

Computers and Chemical Engineering 108 (2018) 289–299

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

MPC-based reference governor control of a continuous stirred-tank reactor

Juraj Holaza*, Martin Klaučo, Ján Drgoňa, Juraj Oravec, Michal Kvasnica, Miroslav Fikar

Institute of Information Engineering, Automation, and Mathematics, Slovak University of Technology in Bratislava, Radlinského 9, SK-812 37, Slovak Republic

ARTICLE INFO

Article history:
Received 1 March 2017
Received in revised form 4 September 2017
Accepted 23 September 2017
Available online 3 October 2017

Keywords:
Model predictive control
Parametric programming
Reference governor
Continuous stirred-tank reactor

ABSTRACT

Optimal control of a CSTR represents a challenging task. The proposed paper discusses two issues. The first one addresses control of pH in a chemical vessel, where the reaction between sodium hydroxide and acetic acid occurs. The objective here is to improve control performance of a well tuned PI controller. It will be shown that this can be achieved by introducing a reference governor scheme. The second problem, elaborated in this paper, is the implementation of the reference governor paradigm. Concretely, we aim to design a fast and cheap MPC-based feedback controller. To achieve these goals, we exploit the region-less explicit technique, which efficiently reduces memory footprint issues of standard explicit MPC schemes. Such MPC-based reference governor was employed to control pH in the chemical vessel. Its control performance is compared with conventional PI controller. Finally, comparison of implementation requirements of region-less and region-based explicit techniques is investigated.

1. Introduction

Maintaining a specific value of pH plays in a wide range of biotechnical and chemical processes. One of the most studied fields is waste water treatment, where the emphasis is on discharging a neutral water into rivers. It has been reported that value of pH has a great impact on the water purification quality (Qian et al., 2014) as bad pH conditions may result in a production of bacteria dangerous to human health. Other important applications can be found, e.g., in coagulation processes (Qin et al., 2006), metal absorption from water (Mandal et al., 2015), microalgae production (Pawlowski et al., 2014), pulp and paper production (Krogell et al., 2015), or medical research and medicine preparation (Georgiev et al., 2013).

With increasing pressure on product quality, process efficiency and environmental protection, well performing pH control is highly desired. This is, however, changeling task mainly due to a strong non-linear relation between pH and concentration of acid/base (King, 2010). To deal with this issue, more complex models were designed to obtain better control performance (Hermansson and Syafiie, 2015). Moreover, numerous advanced control strategies were developed to control the pH processes. Particularly, fuzzy controllers (Fuente et al., 2006; Chen et al., 2011), adaptive controllers (Narayanan et al., 1997; Salehi et al., 2009), or generalized predictive control feedback laws (Altnten, 2007) were employed, to name a few. Even though controller design via conventional means has been proven to be insufficient (Ibrahim, 2008) in several cases, a simple PID controller is still commonly used in pH control.

Nowadays, PID controllers are the most widespread control algorithms as they are used in more than 90% of the overall simple feedback loops (Desborough and Miller, 2002). This control methodology is popular mainly due to its simple structure, robustness, low computational/hardware demands, easy design, and straightforward tuning. On the other hand, the drawback of all PID controllers is that they do not *apriori* consider constraints in the design. Even though, that this deficiency is usually tackled via additional blocks, e.g., such as anti-windup schemes, tasks like *minimization of power consumption* or *specific safety conditions* are not an easy task to include into the control design Åström and Hä (2006).

On the other hand, model predictive control (MPC) excels in providing all of the aforementioned requirements, see Maciejowski (2001). This methodology represents the state-of-the-art control strategy, which finds its application especially in the industrial field (Qin and Badgewell, 2003). However, if PID controllers are already well adopted, the transition from PID into MPC is generally considered to be non-smooth and thus is not preferable, e.g., due to additional investments. Therefore, in this paper, we aim to

* Corresponding author.
E-mail addresses: juraj.holaza@stuba.sk (J. Holaza), martin.klauco@stuba.sk (M. Klaučo), jan.drgona@stuba.sk (J. Drgoňa), juraj.oravec@stuba.sk (J. Oravec), michal.kvasnica@stuba.sk (M. Kvasnica), miroslav.fikar@stuba.sk (M. Fikar).

https://doi.org/10.1016/j.compchemeng.2017.09.020
0098-1354/

exploit MPC policy as an outer optimization layer for, e.g., already employed, PID controllers in order to improve control performance. The main idea here is to shape references to PID controllers in an optimal fashion using MPC. Such an approach is commonly used, e.g., in chemical or petrochemical industry with the objective of increasing the overall production and provide additional safety, i.e., to guarantee constraints satisfaction while minimizing specified performance criterion. In literature, such paradigm of MPC and PID controllers are referred as a *reference governor control* (Borrelli et al., 2009; Bemporad, 1998).

As it was pointed out, implementation of MPC is not trivial since to maintain grantees of optimality, stability and constraints satisfaction, the optimization problem has to be solved within the duration of one sample period. This requirement comes hand in hand with a need of sufficient computational resources in control hardware and an appropriate solver. These deficiencies can be mitigated via introducing explicit MPC, see. Bemporad et al. (2002), Baotić et al. (2008), Pistikopoulos (2012). Here parametric programming (Gal and Nedoma, 1972) is used to offline pre-calculate analytic solution of the MPC optimization problem. The implementation is then reduced only to a simple table searching and subsequent mere function evaluation of an affine expression, both of which can be executed efficiently even in a low-cost control hardware. Furthermore, to mitigate even the memory footprint of such controller, we propose to encode the analytic solution via so-called region-less approach, which was pioneered in Borrelli et al. (2010) and enhanced in Kvasnica et al. (2015), Drgoň et al. (2017).

This work directly extends the results proposed in Holaza (2016), i.e., we aim to control a continuous stirred-tank reactor (CSTR), which consists of two storage tanks, two pumps and a vessel, where a chemical reaction between the solution of acetic acid and the solution of sodium hydroxide takes place. The control objective is to adjust the flow rate of the alkaline stream (while the acidic stream remains uncontrolled) such that pH of the mixed product will track specified reference. To provide a good control performance, we aim to design a reference governor scheme. Here, MPC-based optimization layer shapes optimal references to a conventional PI controller, which controls pH in the chemical vessel via manipulating voltage of the pump feeding the reaction vessel with the alkaline solution. It will be shown that implementation requirements of such control paradigm can be minimized via employing region-less explicit approach.

The paper is organized as follows: experimental setup of the controlled CSTR is introduced in Section 2. Prediction model of CSTR is derived based on the measured experimental data in Section 3. Control setup of the inner closed-loop with PI controller is presented in Section 4, followed by MPC-based reference governor design in Section 5. In Section 6 are discussed the experimental results of the closed-loop control performance, and the main conclusions are formulated in Section 7.

2. Experimental process

This section provides basic description of the experimental process, introduces chemicals, and defines chemical reaction, which occurs inside of the chemical vessel. Moreover, here we state the control setup and the control objective.

2.1. Process equipment and model

The controlled plant represents a laboratory CSTR of Armfield PCT40 and auxiliary equipment. The scheme of the pH process is depicted in Fig. 1. The process consists of two retention tanks that provide solutions of inlet acid of concentration c_A and inlet base of concentration c_B. Solutions are transferred from these tanks into the bottom part of chemical vessel via two identical peristaltic pumps, i.e., p_A for acid and p_B for base. Here both streams are continuously mixed by a stirrer and pH of the effluent is measured by a probe. For convenience, short list of all components along with their descriptions is provided next:

- chemical vessel with volume of $V = 1.5\,\mathrm{dm}^3$,
- two peristaltic pumps (p_A, p_B) each of which admits voltage of (u_A, u_B) within [0, 5] V, what correspond to the flow rates (f_A, f_B) of [0, 10] ml s^{-1},
- two storage tanks that can hold up to 100 dm^3 of solution each,
- pH probe which returns signal in range of [0, 14].

We consider the acetic acid (CH_3COOH) and the sodium hydroxide (NaOH), with concentration of solutes $c_A = 0.01\,\mathrm{mol\,m}^{-3}$ and $c_B = 0.01\,\mathrm{mol\,m}^{-3}$, respectively. By dissolving aforementioned solutes in water, following dissociation reactions take place

$$\mathrm{NaOH} \overset{\mathrm{H_2O}}{\rightarrow} \mathrm{Na+} + \mathrm{OH}^-, \tag{1a}$$

$$\mathrm{HAc} \overset{\mathrm{H_2O}}{\rightleftharpoons} \mathrm{H}^+ + \mathrm{Ac}^-, \tag{1b}$$

where Na^+ is sodium cation, OH^- is hydroxide, H^+ is hydrogen cation, HAc is abbreviation of acetic acid and Ac^- is acetate (ethanoate) ion CH_3COO^-. Sodium hydroxide is a strong base that dissolves in the water completely while acetic acid is a weak acid that dissociates only partially, what means that besides both ions Ac^- and H^+ there exist molecules of HAc. The expression $[\cdot]$ denotes the concentration of a specific chemical, i.e., $[OH^-]$ stands for the concentration of hydroxide in $\mathrm{mol\,m}^{-3}$.

The pH of the base solution with concentration $c_B = 0.01\,\mathrm{mol\,m}^{-3}$ can be computed from

$$\mathrm{pH} = -\log_{10}[\mathrm{H}^+], \tag{2}$$

where concentration of hydrogen cation can be determined per dissociation constant of water

$$k_w = [\mathrm{OH}^-][\mathrm{H}^+] \approx 10^{-14}. \tag{3}$$

Therefore, by plugging (3) into (2), the pH of the sodium hydroxide solution is $\mathrm{pH} = -\log_{10} k_w + \log_{10}[\mathrm{OH}^-] = 12$. On the other hand, in order to compute pH of the acetic acid we need to introduce dissociation constant of the acetic acid

$$k_a = \frac{[\mathrm{H}^+][\mathrm{Ac}^-]}{[\mathrm{HAc}]} \approx 10^{-5}. \tag{4}$$

Its pH value can be then computed as a positive root of the following quadratic equation

$$x^2 + k_a x - c_A k_a = 0, \tag{5}$$

thus pH of the acetic acid with concentration $c_A = 0.01\,\mathrm{mol\,m}^{-3}$ is 3.5.

Electro-neutrality of the solution gives

$$[\mathrm{Na}^+] + [\mathrm{H}^+] = [\mathrm{OH}^-] + [\mathrm{Ac}^-]. \tag{6}$$

The material balance of the entire system is given as

$$V\frac{\mathrm{d}x_1}{\mathrm{d}t} = f_A c_A - (f_A + f_B)x_1, \tag{7a}$$

$$V\frac{\mathrm{d}x_2}{\mathrm{d}t} = f_B c_B - (f_A + f_B)x_2. \tag{7b}$$

where x_1 is the concentration of acetic acid. Specifically, based on (1b), $x_1 = [\mathrm{HAc}] + [\mathrm{Ac}^-]$. State x_2 is the concentration of sodium cation, namely $x_2 = [\mathrm{Na}^+]$. Next, by plugging $[\mathrm{HAc}] = x_1 - [\mathrm{Ac}^-]$ into (4) we obtain

$$[\mathrm{Ac}^-] = \frac{x_1 k_a}{k_a + [\mathrm{H}^+]}, \tag{8}$$

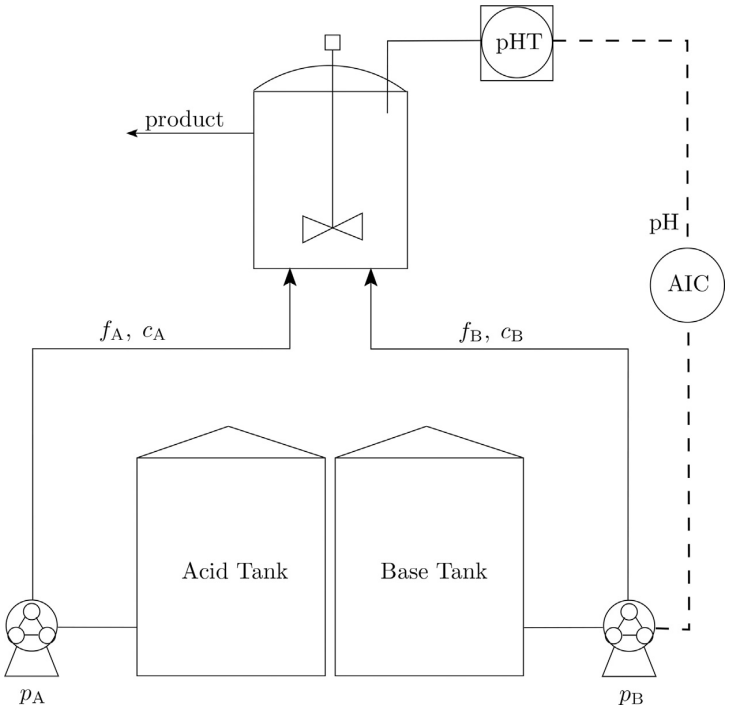


Fig. 1. Illustration of neutralization reaction vessel, where controlled variable is pH and control variable is voltage to the pump P_B.

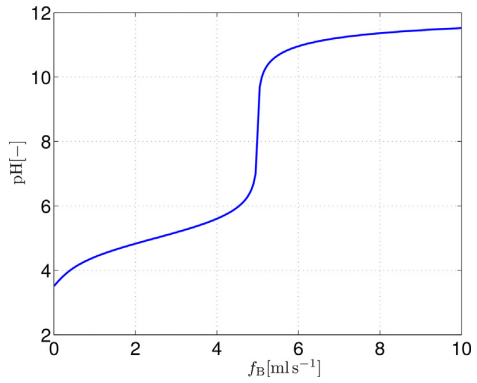


Fig. 2. Titration curve of the sodium hydroxide and the acetic acid, with $f_A = 5$ ml s^{-1}.

which, together with (8), can substitute equilibrium concentrations in (6) resulting in

$$x_2 + [H^+] = \frac{k_w}{[H^+]} + \frac{x_1 k_a}{k_a + [H^+]}. \tag{9}$$

After straightforward manipulations (9) can be rewritten into algebraic cubic equation, given as

$$[H^+]^3 + x_2 + k_a[H^+]^2 + x_2 k_a - x_1 k_a - k_w[H^+] - k_w k_a = 0. \tag{10}$$

The complete model of the considered pH neutralization process is then given by two differential equations (7) and one cubic algebraic equation (10). The solution of (10) is known as a titration curve and is shown in Fig. 2. We can observe the non-linear nature of the process, which is characterized by S-shape curve. To design a control strategy not only for the full range of pH but even for a specific interval is a non-trivial task. Therefore, an advanced optimization-based control strategy needs to be designed to ensure the high quality of the closed-loop control performance.

2.2. Problem statement

Consider process described in Section 2.1, where streams of the acid solution and the base solution are pumped into the chemical vessel via p_A and p_B, respectively. Both streams are here mixed with the stirrer and the final product is then taken from the top of the chemical vessel such that volume of the mixture remains the same. The pH probe is placed right next to the outgoing mixture and the flow rate of the acidic solution is maintained constant, i.e., derivatives $\dot{f}_A = \dot{u}_A = 0$. The control objective is to manipulate the voltage u_B of the pump p_B in such manner that the pH of the mixed product will attain the desired reference. The list of all variables is summarized next:

1. controlled variable is the pH of the stream from the chemical vessel, $\bar{y} \in \mathbb{R}$, where $\bar{y} \in [0, 14]$ pH,
2. manipulated variable (input) is the voltage in the pump p_B, $\bar{u}{:=}u_B \in \mathbb{R}$, with $\bar{u} \in [0, 5]$ V,
3. uncontrolled variables:
   - voltage in the pump p_A, $u_A = 2.5$ V ($f_A \approx 5$ ml s^{-1}),
   - concentration of the acetic acid solution: $c_A = 0.01$ M,
   - concentration of the sodium hydroxide solution: $c_B = 0.01$ M.

The main goal is to design such a control strategy that optimizes the closed-loop control performance subject to the nonlinear CSTR behavior in various working points taking into account constraints on control and manipulated variables. Moreover, it is required that the overall computational demand of the implemented strategy is minimized.

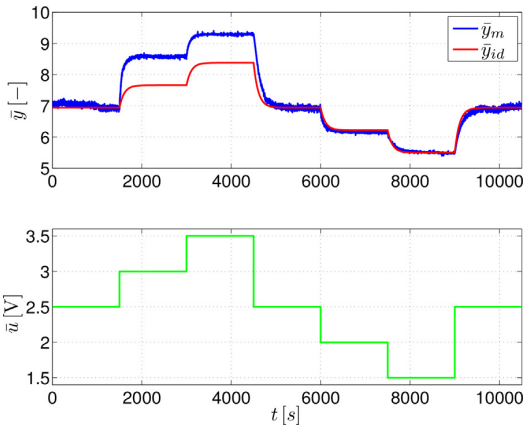


Fig. 3. Experimental validation of the identified model of CSTR. (For interpretation of the references to color in text, the reader is referred to the web version of the article.)

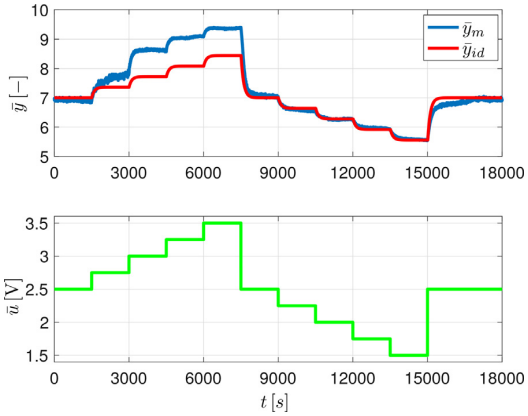


Fig. 4. Model verification against new data set.

3. Experimental model of the controlled process

The objective of this section is to derive a mathematical model of the controlled process. This is a crucial task as an accurate model represents a cornerstone for all model-based control design techniques. In the context of this paper, the design model will be used to derive PI controller, and as a prediction model in MPC. Even though that there exist several types of models (e.g. linear, nonlinear, or piecewise affine), we need to keep in mind that in this case accuracy of the model comes hand in hand with its complexity. And since we aim to design a control strategy with low implementation requirements, i.e., the control algorithm will be easily implementable also in standard control platforms with limited computation and memory resources, we restrict ourselves for a single state linear time-invariant (LTI) model of the form

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{11a}$$

$$y(t) = Cx(t), \tag{11b}$$

where $x(t) \in \mathbb{R}$, $u(t) \in \mathbb{R}$, $y(t) \in \mathbb{R}$ denote state, input and output, respectively, at time t. A, B, C are state space coefficients and $\dot{x}(t) \in \mathbb{R}$ is the state derivative at time t. We distinguish all real-valued variables from their deviation form with bar. Concretely, $\bar{u} = u + \bar{u}^s$ is the real input, $\bar{y} = y + \bar{y}^s$ is the real output, u is the input in deviation form, and y is the output in deviation form. Variables $\bar{u}^s$ and $\bar{y}^s$ denote steady values of input and output, respectively.

To begin the process identification, we have firstly selected operating point $\bar{y}^s = 7$, which approximately correspondents to the steady state control action $\bar{u}^s = 2.5$ V. This point was chosen due to the fact that pH = 7 represents the neutral value that is particularly required in neutralization processes. Moreover, a neighborhood of this point is harmless for a majority of industrial components. To describe the dynamical behavior of the controlled process by (11) around the steady-state output value $\bar{y}^s$, we have chosen step-changes of control actions from the interval $\bar{u}^s \pm 1$ V, concretely

$$u \in \{2.5, 3.0, 3.5, 2.5, 2.0, 1.5\}\ V, \tag{12}$$

with step change period 1500 s.

The corresponding input and output profiles are depicted in Fig. 3 by green and blue solid lines. To suppress the high-frequency noise of the measured data the first order Butterworth filter was used with cut-off frequency 0.01 rad s^{-1}. Based on the measured data, the controlled system was identified by the first order system (11). The identified dynamics in the continuous time domain are in form of

$$\dot{x}(t) = -0.9524 \times 10^{-2} x(t) + 0.1250\ u(t), \tag{13a}$$

$$y(t) = 0.2196\ x(t). \tag{13b}$$

The same sequence of control inputs (12) was used to verify the identified model (13). The corresponding output profile is depicted in Fig. 3 and it is denoted by the solid red curve. It can be observed that the controlled process is nonlinear as it exhibits non-symmetrical responses. Therefore, while the precision of the identified model tracks the acidic environment (pH < 7) well, the precision in the alkaline environment (pH > 7) is not so accurate. The model (13) was also verified against a new set of input–output data from the process (see Fig. 4). Here we can observe the same relation between the model and process, i.e., the model is accurate in the acidic environment while not so good near pH = 8. This model mismatch will be compensated by the designed controller.

4. Inner closed-loop with PI controller

A simple closed-loop control scheme is proposed as a baseline control strategy against which the MPC-base controller will be evaluated. Such a scheme is depicted in Fig. 5, where the PI controller is implemented with back-calculation anti-windup method as visualized in Fig. 6. Here, a PI controller must be designed in order to fulfill the control objectives. Family of PID controllers is widely used in every branch of the industry, pH control included. The main reason why such controllers are still the first choice in industry application is their simple structure, robustness, and very low computational demands.

There are numerous methods on tuning of a PID controller, the majority of which is already embedded in various toolboxes, see e.g. Oravec and Bakošová (2012), The MathWorks (2016). In this paper, we have used interactive toolbox (Oravec and Bakošová, 2012) to design PI controller (14) in its standard structure. Proportional gain $P = 1.750$ and integral gain constant $I = 0.0167$ (see Fig. 6) of the controller were chosen such that closed-loop response was fast and without overshoots. The input–output formulation of such PI controller is given by

$$\dot{u}(t) = 1.7500\dot{e}(t) + 0.0292e(t),$$

what can be translated into a state-space representation as

$$\dot{x}_{\mathrm{PI}}(t) = 0.1250\ e(t), \tag{14a}$$

$$u(t) = 0.2333\ x_{\mathrm{PI}}(t) + 1.7500\ e(t), \tag{14b}$$

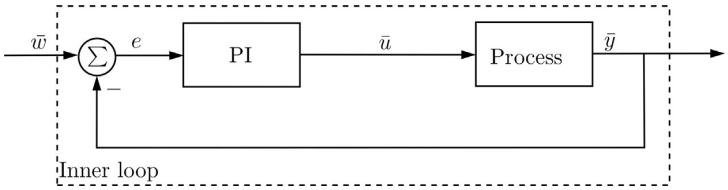


Fig. 5. Closed-loop scheme with PI controller.

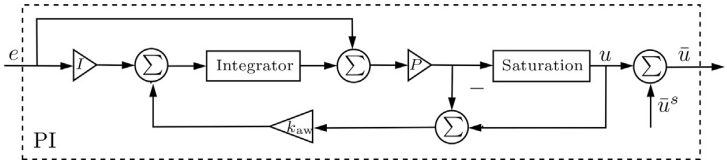


Fig. 6. PI controller scheme with back-calculation anti-windup method. Recall that the manipulated variable u is the voltage on the pump p_B.

where $x_{PI}(t)$, $e(t)$ is the controller state and tracking error at time t, respectively. Due to the presence of constraints on control inputs, anti-windup technique is considered with back-calculation coefficient $k_{aw} = 1.1711$, see Fig. 6. The variable k_{aw} directly influences the performance of the anti-windup scheme. Increasing this variable will result in decreased rise time of the control performance.

5. MPC-based reference governor

This section elaborates construction of MPC-based reference governor as an outer control layer for the inner closed-loop, which consists of PI controller (14) and the controlled system. The objective of the reference governor is to shape the reference w for the inner control loop, such that the pH of the outgoing product $\bar{y}$ will track the user specified reference $\bar{r}$, while satisfying all of the predefined constants. Such control scheme is depicted in Fig. 7. Note that all variables with bar are real values, otherwise they are in their deviation form. In the sequel, we start by designing a prediction model for the MPC-based reference governor, then we introduce a mathematical formulation of such control scheme. Finally, we compute its explicit (analytic) solution and discuss its complexity.

5.1. Prediction model

Since MPC is a model-based control strategy, one need to design an appropriate prediction model. In this case, such prediction model has to include the entire inner loop of our control scheme shown in Fig. 5. This is actually quite an easy task, as we already have both identified the model of the controlled process (13) and PI algorithm (14) in hand. To proceed, we firstly need to specify sampling period $\Delta = 5$ s. This value was chosen based on the dynamics of the closed-loop system. Subsequently, by exploiting Δ, $e(t) = \bar{w}(t) - \bar{y}(t)$ and straightforward mathematical manipulations (Klaučo and Kvasnica, 2017), one obtains the discrete-time state-space model of the inner loop:

$$\mathbf{x}(t+\Delta) = \bar{\mathbf{A}}\mathbf{x}(t) + \bar{\mathbf{B}}w(t), \tag{15a}$$

$$u(t) = \bar{\mathbf{C}}_u\mathbf{x}(t) + \bar{\mathbf{D}}_u w(t), \tag{15b}$$

$$y(t) = \bar{\mathbf{C}}_y\mathbf{x}(t), \tag{15c}$$

with

$$\bar{\mathbf{A}} = \begin{bmatrix} 0.9909 & -0.1189 \\ 0.1263 & 0.7416 \end{bmatrix}, \ \bar{\mathbf{B}} = \begin{bmatrix} 0.5548 \\ 0.9887 \end{bmatrix}, \ \bar{\mathbf{C}}_u = \begin{bmatrix} 0.2333 & -0.3844 \end{bmatrix}, \ \bar{\mathbf{D}}_u = \begin{bmatrix} 1.7500 \end{bmatrix}, \ \bar{\mathbf{C}}_y = \begin{bmatrix} 0.0000 & 0.2196 \end{bmatrix}.$$

The prediction model does not precisely fit the dynamics of the combined plant and controller. To suppress this model mismatch one needs to employ offset-free tracking (Borrelli and Morari, 2007; Muske and Badgwell, 2002; Pannocchia and Rawlings, 2003) to track the controlled output to the desired reference. This approach is based on extending the model (15) by disturbance d, which is added to the output equation (15c). Such augmented model is defined as:

$$\mathbf{x}(t+\Delta) = \bar{\mathbf{A}}x(t) + \bar{\mathbf{B}}w(t), \tag{16a}$$

$$u(t) = \bar{\mathbf{C}}_u x(t) + \bar{\mathbf{D}}_u w(t), \tag{16b}$$

$$y(t) = \bar{\mathbf{C}}_y x(t) + d(t), \tag{16c}$$

$$d(t+\Delta) = d(t). \tag{16d}$$

Without loss of generality, we assume in (16d) that the disturbance d is considered constant, as we implement MPC in well-known moving horizon control fashion. The state and disturbance observer for the system (16) can be given as:

$$\begin{bmatrix} \hat{\mathrm{x}}(t+\Delta) \\ \hat{d}(t+\Delta) \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{A}} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathrm{x}}(t) \\ \hat{d}(t) \end{bmatrix} + \begin{bmatrix} \bar{\mathbf{B}} \\ 0 \end{bmatrix} (\bar{u}(t) - \bar{u}^s) + \hat{y}(t), \tag{17a}$$

$$\hat{y}(t) = \begin{bmatrix} \mathrm{k}_1 \\ k_2 \end{bmatrix} (\bar{y}(t) - \bar{y}^s) - \bar{\mathbf{C}}_y\hat{x}(t) - \hat{d}(t), \tag{17b}$$

where $\hat{\mathrm{x}}(t)$, $\hat{d}(t)$, $\hat{y}(t)$ denote estimates of states, disturbance and output, respectively, $\bar{y}^s$ is the steady output and $\bar{y}(t)$ is the output at time t. Gains $\mathbf{k}_1$ and k_2 are chosen such that the observer is stable and further tuned to achieve desirable convergence. In our setup, we have used $\mathbf{k}_1 = [0.01880.0217]^\top$ and $k_2 = 0.2693$.

5.2. MPC formulation

To improve control performance of the PI controller (14), we aim to design a reference governor scheme, which will enforce constraints of the controlled variable $\bar{y}$. Yet, these constraints will be softened by slack variables s. This way, we can introduce an additional (more penalized) pH interval, which we would like to avoid. Moreover, the feasibility of such feedback law will be enforced. The MPC optimization problem of such reference governor scheme can be stated as:

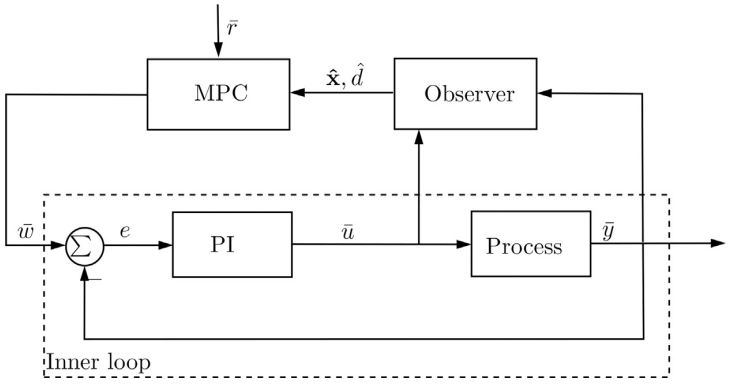


Fig. 7. Closed-loop scheme with the reference governor.

$$\min_{\mathbf{w}} \sum_{k=0}^{N-1} \left(q_s s^2 + q_w (w_k - w_{k-1})^2 + q_y (y_k + \bar{y}^s) - \bar{r}^2 \right) \tag{18a}$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \bar{\mathbf{A}}\mathbf{x}_k + \bar{\mathbf{B}} w_k, \ \forall k \in \mathbb{N}_0^{N-1}, \tag{18b}$$

$$u_k = \bar{\mathbf{C}}_{\mathrm{u}}\mathbf{x}_k + \bar{\mathbf{D}}_{\mathrm{u}} w_k, \ \forall k \in \mathbb{N}_0^{N-1}, \tag{18c}$$

$$y_k = \bar{\mathbf{C}}_{\mathrm{y}}\mathbf{x}_k + d_0, \ \forall k \in \mathbb{N}_0^{N-1}, \tag{18d}$$

$$\mathbf{x}_0 = \hat{\mathrm{x}}(t), \ d_0 = \hat{d}(t), \tag{18e}$$

$$s \geq 0, \tag{18f}$$

$$u_{\min} \leq u_k \leq u_{\max}, \ \forall k \in \mathbb{N}_0^{N-1}, \tag{18g}$$

$$w_{\min} \leq w_k \leq w_{\max}, \ \forall k \in \mathbb{N}_0^{N-1}, \tag{18h}$$

$$y_{\min} - s \leq y_k \leq y_{\max} + s, \ \forall k \in \mathbb{N}_0^{N-1}, \tag{18i}$$

where $\mathbb{N}_a^b$ is a set of positive integers, i.e., $\mathbb{N}_a^b = \{a, a+1, a+2, \ldots, b\}$. Denote $\mathbf{x}_k \in \mathbb{R}^2$, $w_k \in \mathbb{R}$, $u_k \in \mathbb{R}$ and $y_k \in \mathbb{R}$ to be states of the inner loop, the shaped reference, the control input from the PI controller and the output from the process at the kth prediction step, respectively. Next, $N \in \mathbb{N}$ is prediction horizon, $s \in \mathbb{R}$ is slack variable, $\bar{r} \in \mathbb{R}$ is desired reference, w_{k-1} is the shaped reference form the previous sample instant and q_s, q_w, q_y are positive definite weighting coefficients. Moreover, $\hat{d}(t)$ and $\hat{\mathrm{x}}(t)$ are estimated disturbances and states, respectively, given by the observer (17). The disturbance is considered to be constant along the prediction horizon.

Optimization problem in (18) is initialized, at each discrete time t, by the vector of parameters $\boldsymbol{\theta}(t) := [\hat{\mathrm{x}}(t)^\top, \hat{d}(t), w*(t-\Delta), \bar{r}(t)]^\top \in \mathbb{R}^5$, where, e.g., $w*(t-\Delta)$ denotes optimal value of the shaped reference used in the previous sample instant Δ. The vector of optimized variables is defined by $\mathbf{w} := [w_0, \ldots, w_{N_c-1}, s]^\top \in \mathbb{R}^{N+1}$.

Each part of the objective function and every constraint in (18) has the following impact on the optimization problem. In (18a) the first two terms penalize slack variable and fluctuations of the shaped references, while the last term provides convergence of the process output to the desired reference. The dynamics of the inner loop is described by (18b)–(18d), (18e) represents initial conditions of the optimization problem, (18f) subjects slack variable to attain only non-negative numbers, and constraints (18g)–(18i) restrict values of control inputs, shaped references, and controlled outputs, respectively.

The closed-loop implementation of the MPC-based reference governor in (18) can be executed, with the receding horizon fashion, via following steps:

1. choose an initial vector $\boldsymbol{\theta}(t)$,
2. solve optimization problem (18) with $\boldsymbol{\theta}(t)$, what yields vector of optimized variables $\mathbf{w}* := [w_0*, \ldots, w_{N-1}*, s*]^\top$,
3. extract the first reference $w(t)* = w_0*$ and sent it to the inner loop,
4. based on the process output $y = \bar{y} - \bar{y}^s$ and applied control input $w(t)*$, use (17) to observe states $\hat{\mathrm{x}}(t+\Delta)$ and disturbance $\hat{d}(t+\Delta)$,
5. at next sample period construct new parameter vector $\boldsymbol{\theta}(t+\Delta) = [\hat{\mathrm{x}}(t+\Delta), \hat{d}(t+\Delta), w(t)*, \bar{r}(t+\Delta)]^\top$ and go back to the second step.

Since the objective function (18a) is quadratic and all constraints (18b)–(18i) are linear, the optimization problem (18) is a quadratic program (QP), which can be solved be numerous state-of-the-art solvers such as Gurobi Optimization (2014) or ILOG (Inc., 2003). The problem, however, is that even with appropriate solver in hand, which generally is on its own expensive, solving (18) at each sample instant Δ can be computationally demanding, i.e., power, time, and/or hardware issues. Therefore, in the sequel, we will show how to avoid these requirements, what will lead to a much simpler MPC implementation.

5.3. Formulation of region-less explicit MPC

In this section, we will compute an analytic solution of QP in (5.2). Consider $\mathbf{w}$ to be a vector of optimized variables and $\boldsymbol{\theta}$ to be a vector of free parameters, initial conditions of (5.2), respectively. After straightforward mathematical manipulations, the MPC optimization problem (5.2) can be converted into

$$\min_{\mathbf{w}} \ \mathbf{w}^\top \mathbf{H}\mathbf{w} + \boldsymbol{\theta}^\top \mathbf{F}\mathbf{w} \tag{19a}$$

$$\text{s.t.} \ \mathbf{G}\mathbf{w} \leq \mathbf{e} + \mathbf{S}\boldsymbol{\theta}, \tag{19b}$$

what is a parametric quadratic program (pQP).

As it was reported, e.g., in Borrelli (2003), Baotić et al. (2008), the analytic solution of (19) is a continuous piecewise affine (PWA) function, which maps vectors from the parameter space into the

space of optimal control actions, i.e., $\kappa(\boldsymbol{\theta}): \boldsymbol{\theta} \rightarrow \mathbf{w}^*$. Such function can be defined as

$$\mathbf{w}* = \kappa(\boldsymbol{\theta}) := \begin{cases} \mathbf{f}_1\boldsymbol{\theta} + f_1 & \text{if } \boldsymbol{\theta} \in \mathcal{R}_1, \\ \vdots & \\ \mathbf{f}_R\boldsymbol{\theta} + f_R & \text{if } \boldsymbol{\theta} \in \mathcal{R}_R, \end{cases} \tag{20}$$

where $\mathcal{R}_i$ denote ith critical polyhedral region, $\mathbf{f}_i$ and f_i are local affine expressions for all $i \in \mathbb{N}_1^R$. Furthermore, for each critical region holds:

1. $\text{int}(\mathcal{R}_i) \cap \text{int}(\mathcal{R}_j) = \emptyset, \forall i \neq j$,
2. $\cup_i \mathcal{R}_i = \Omega$,
3. $\Omega = \{\boldsymbol{\theta} \mid \exists \mathbf{w}^* : \mathbf{G}\mathbf{w}^* \leq \mathbf{e} + \mathbf{S}\boldsymbol{\theta}\}$,

with $\text{int}(\mathcal{R}_i)$ denoting interior of the ith critical region. Subsequently, with $\kappa(\boldsymbol{\theta})$ in hand, the online evaluation of optimal control actions $\mathbf{w}^*$ is reduced only to a mere and division free function evaluation, i.e., consisting only of addition and multiplication. Needlessly to say, both solutions of $\kappa(\boldsymbol{\theta})$ and (18) return the same results.

The shortcoming, however, of (20) lies in the memory demands. This is due to the fact that in order to compute $\mathbf{w}^*$ one need to store Ω and all affine expressions $\mathbf{f}_i$ and f_i, respectively. And it should be stress out that by increasing complexity of (18), i.e., by increasing the number of optimized variables or the number of constraints, the number of critical regions can grow exponentially, thus the memory requirement of (20) that can easily exceed available resources of the targeted control hardware. To tackle this issue, so called region-less explicit MPC was introduced.

Contrary to the standard (region-based) explicit MPC, the region-less approach suggests expressing the analytic solution of (19) without storing critical regions as polyhedral objects. Instead, this methodology encodes the analytic solution by exploiting only matrices of the original pQP (19) and the set of optimal active set combinations $\mathcal{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_R\}$, where $\mathcal{A}_i \subseteq \{1, \ldots, n_c\}$ denotes set of locally active constraints, i.e., where (19b) holds with equality, and n_c denotes the number of overall constraints. The main advantage coming out from this methodology is significant memory savings, what extends the applicability of this control strategy even for large scale systems.

To compute such region-less solution one needs to follow two-step procedure, which can be performed completely offline. Both steps rely on Karush–Kuhn–Tucker (KKT) conditions of the pQP (19) which can be formulated as:

$$\mathbf{G}_{\mathcal{A}_i}\mathbf{w}* = \mathbf{e}_{\mathcal{A}_i} + \mathbf{S}_{\mathcal{A}_i}\boldsymbol{\theta}, \tag{21a}$$

$$\mathbf{G}_{\mathcal{N}_i}\mathbf{w}* < \mathbf{e}_{\mathcal{N}_i} + \mathbf{S}_{\mathcal{N}_i}\boldsymbol{\theta}, \tag{21b}$$

$$\boldsymbol{\lambda}* \geq \mathbf{0}, \tag{21c}$$

$$\mathbf{H}\mathbf{w}* + \mathbf{F}^\top\boldsymbol{\theta} + \mathbf{G}_{\mathcal{A}_i}\boldsymbol{\lambda}* = \mathbf{0}, \tag{21d}$$

$$\boldsymbol{\lambda}_{\mathcal{A}_i}{}^\top(\mathbf{G}_{\mathcal{A}_i}\mathbf{w}* - \mathbf{e}_{\mathcal{A}_i} - \mathbf{S}_{\mathcal{A}_i}\boldsymbol{\theta}) = \mathbf{0}, \tag{21e}$$

where (21a) with (21b) denote primal feasibility, (21c) denote dual feasibility, (21d) optimality and (21e) complementary slackness conditions, respectively. Moreover, $\boldsymbol{\lambda}$ is vector of the dual variables, $\mathbf{G}_{\mathcal{A}_i}$ is a matrix obtained by taking rows of $\mathbf{G}$ based on indices in $\mathcal{A}_i$. The same notation is applied also for matrices $\mathbf{e}_{\mathcal{A}_i}$, $\mathbf{S}_{\mathcal{A}_i}$, $\mathbf{G}_{\mathcal{N}_i}$, $\mathbf{e}_{\mathcal{N}_i}$ and $\mathbf{S}_{\mathcal{N}_i}$, with $\mathcal{N}_i$ denoting set of non-active local constraints, i.e., $\mathcal{N}_i = \{1, \ldots, n_c\} \backslash \mathcal{A}_i$.

In the first step, the set of all locally optimal active sets combinations $\mathcal{A}$ is enumerated. Generally, this means that one needs to solve (21) for each possible active set candidate, the total number of which is given by

$$r_{\max} = \sum_{k=0}^{|\mathbf{w}|} \frac{n_c!}{k!(n_c-k)!}, \tag{22}$$

where $|\mathbf{w}|$ denotes the number of optimization variables. The problem, however, is that $r_{\max}$ grows exponentially with the number of optimized variables, thus this approach becomes quickly unpractical. The remedy for this problem was proposed in Gupta et al. (2011), where authors suggested an efficient branch-and-bound method how to explore active set candidates. Here, additional feasibility checks are performed in order to cut off redundant branches in an exploration tree. This way exploration complexity is significantly reduced, with the upper bound equal to $r_{\max}$.

Next, in the second step, we construct local affine maps $\boldsymbol{\theta} \rightarrow \mathbf{w}^*$ for all optimal active set combination in $\mathcal{A}$. Concretely, the vector of optimal variables can be formulated from (21) as

$$\mathbf{w}* = -\mathbf{H}^{-1}(\mathbf{F}^\top\boldsymbol{\theta} + \mathbf{G}_{\mathcal{A}_i}^\top\boldsymbol{\lambda}*), \tag{23}$$

where we assume that the Hessian $\mathbf{H}$ is invertible.[1] Furthermore, an affine expression of the optimal dual variables can be obtained by plugging (23) into (21a) what leads to

$$\boldsymbol{\lambda}* = \mathbf{Q}(\mathcal{A}_i)\boldsymbol{\theta} + \mathbf{q}(\mathcal{A}_i), \tag{24}$$

where[2]

$$\mathbf{Q}(\mathcal{A}_i) = -(\mathbf{G}_{\mathcal{A}_i}\mathbf{H}^{-1}\mathbf{G}_{\mathcal{A}_i}^\top)^{-1}(\mathbf{S}_{\mathcal{A}_i} + \mathbf{G}_{\mathcal{A}_i}\mathbf{H}^{-1}\mathbf{F}^\top), \tag{25a}$$

$$\mathbf{q}(\mathcal{A}_i) = -(\mathbf{G}_{\mathcal{A}_i}\mathbf{H}^{-1}\mathbf{G}_{\mathcal{A}_i}^\top)^{-1}\mathbf{e}_{\mathcal{A}_i}. \tag{25b}$$

Finally, with pQP data $\mathbf{H}, \mathbf{F}, \mathbf{G}, \mathbf{e}, \mathbf{S}$ of (19), the list of all active sets $\mathcal{A}$ and $\mathbf{Q}(\mathcal{A}_i)$ with $\mathbf{q}(\mathcal{A}_i)$ for all $i \in \mathbb{N}_1^R$ in hand,[3] online evaluation of optimal control actions $\mathbf{w}^*$ is reduced only to a simple linear algebra. Specifically, for a given initial condition $\boldsymbol{\theta}$ one needs to find an index $i \in \mathbb{N}_1^R$ for which both the dual feasibility (21c) as well as the primal feasibility (19b) hold at the same time, i.e.,

$$\boldsymbol{\lambda}* \geq \mathbf{0}, \tag{26a}$$

$$\mathbf{G}\mathbf{w}* \leq \mathbf{e} + \mathbf{S}\boldsymbol{\theta}, \tag{26b}$$

with $\boldsymbol{\lambda}^*$ and $\mathbf{w}^*$ given per (24) and (23), respectively. This procedure can be done sequentially as reported in Algorithm 1.

Algorithm 1. Online evaluation of region-less explicit MPC.

$\mathcal{A}$, $\mathbf{Q}(\mathcal{A}_i)$, $\mathbf{q}(\mathcal{A}_i)$Require: and pQP matrices H−1, F, G, e, S
function: REGIONLESSMPC$\boldsymbol{\theta}$
$i \in \mathbb{N}_1^R$ 2: for
$\boldsymbol{\lambda} = \mathbf{Q}(\mathcal{A}_i)\boldsymbol{\theta} + \mathbf{q}(\mathcal{A}_i)$ 3: Compute
if $\boldsymbol{\lambda} \geq 0$
$\mathbf{w} = -\mathbf{H}^{-1}(\mathbf{F}^\top\boldsymbol{\theta} + \mathbf{G}_{\mathcal{A}_i}^\top\boldsymbol{\lambda})$ 5: Compute
if $\mathbf{G}\mathbf{w} \leq \mathbf{e} + \mathbf{S}\boldsymbol{\theta}$
$\mathbf{w}^* \longleftarrow \mathbf{w}$
return $\mathbf{w}^*$
end if
end if
end for
end function

Remark 1. An alternative region-less explicit MPC approach was also proposed by Borrelli et al. (2010) where the authors suggest

[1] The Hessian $\mathbf{H}$ in (19) is positive definite, symmetric and invertible if $q_w > 0$ and $q_y > 0$ in (18) holds.

[2] Here, for sake of simplicity, we do not consider the degeneracy of $\mathbf{G}_{\mathcal{A}_i}$, i.e., we assume that $\mathbf{G}_{\mathcal{A}_i}$ is of full row rank. Approaches to tackle degenerate cases can be found, e.g., in Spjø et al. (2005).

[3] We recall that the inverse of the Hessian, i.e., $\mathbf{H}^{-1}$, as well as expressions in (23) and (24) can be pre-computed offline for all $i \in \mathbb{N}_1^R$.

to replace the computation of the primal optimizer in (23) by a complete affine parameterization of the form

$$\mathbf{w} = \mathbf{Q}_{\mathrm{w}}(\mathcal{A}_i)\boldsymbol{\theta} + \mathbf{q}_{\mathrm{w}}(\mathcal{A}_i). \tag{27}$$

As a consequence, one requires more memory to store the factors $\mathbf{Q}_{\mathrm{w}}(\mathcal{A}_i)$ and $\mathbf{q}_{\mathrm{w}}(\mathcal{A}_i)$. However, the upside is that the evaluation of the primal optimizer via (27) is faster compared to the evaluation via (23). The adaptation of Algorithm 1 to the parametric representation of the primal optimiser amounts to replacing Step 5 by (27). Practical consequences of this change will be investigated in the subsequent section.

6. Results and discussion

This section elaborates experimental results of pH control in the chemical reactor. Firstly we specify parameters of MPC setup followed by complexity discussion of its region-less explicit formulation. Subsequently, we present experimental data of PI controller as well as region-less explicit MPC. Control performances of both controllers are compared by means of various criteria.

6.1. Setup of the MPC design

The setup of the MPC policy in (18) was designed as follows. The weighting coefficients were chosen as to equivalently penalize fluctuation of the shaped reference and the reference tracking, while the emphasis was put into the penalty of slack variables, i.e., $q_s = 10^3$, $q_w = 10$, $q_y = 10$. Prediction model (18b)–(18d) was given per (16) and initialized by the vector of parameters in the form $\boldsymbol{\theta}(t) := [\hat{\mathbf{x}}(t), \hat{d}(t), w*(t-\Delta), \bar{r}(t)]^\top \in \mathbb{R}^5$. The prediction horizon of length of 25 s was used, i.e., $N = 5$. To allow MPC to operate only with the admissible input voltage range $\bar{u} \in [0, 5]$ V, the upper and lower bound in (18g) were chosen as $u_{\min} = -2.5$ V and $u_{\max} = 2.5$ V. The shaped reference w was restricted to attain only physically realizable values by setting $w_{\min} = -7$ and $w_{\max} = 7$, what translates into interval of pH $\in [0, 14]$. The output constraints (18i) were $y_{\min} = -1.25$ and $y_{\max} = 1.25$, yet soften by slack variables s due to measuring noise and considered process-model mismatch, see Fig. 3.

With the chosen MPC setup, optimization problem (18) was converted into pQP (19) with 31 constraints. We have employed method proposed in Gupta et al. (2011) to generate the list $\mathcal{A}$, which consisted of 2979 optimal active set combinations, out of $r_{\max} = 942\,648$ possible candidates. For each ith entry of this set, we have pre-factorized expressions $\mathbf{Q}(\mathcal{A}_i)$ and $\mathbf{q}(\mathcal{A}_i)$ per (25a) and (25b), respectively. Both of these steps were carried out in 208 s by using MATLAB R2016b, Multi-Parametric Toolbox (Herceg et al., 2013) and 3.4 GHz processor.

With the pre-computed region-less explicit MPC solution in hand, a self-contained file was generated, which maps the parameter vector $\boldsymbol{\theta}$ onto the vector of shaped references $\mathbf{w}^*$ per Algorithm 1. Specifically, we had to store 438 floating numbers for pQP matrices $\mathbf{H}^{-1}$, $\mathbf{F}$, $\mathbf{G}$, $\mathbf{e}$ and $\mathbf{S}$, 13,430 integer numbers for $\mathcal{A}$, and 80,580 floating numbers for $\mathbf{Q}(\mathcal{A}_i)$ and $\mathbf{q}(\mathcal{A}_i)$. By allocating 2 bytes per integer and 8 bytes per single floating number, the overall memory footprint of the region-less explicit controller is 675 kb when using double-precision arithmetics.

We have also investigated how a complete parameterization of the primal optimizer as in (27) in the spirit of Remark 1 influences the complexity of the region-less solution. Specifically, computing the factors $\mathbf{Q}_{\mathrm{w}}(\mathcal{A}_i)$ and $\mathbf{q}_{\mathrm{w}}(\mathcal{A}_i)$ in (27) increased the off-line construction time by 1 s from 208 s to 209 s. More important, however, is that the storage of the aforementioned factors resulted in a considerable increase of the memory footprint from 675 kb to 1505 kb, confirming the conclusions of Remark 1.

Moreover, we have also applied the standard region-based approach to construct the explicit solution. This task was also performed on 3.4 GHz machine running Multi-Parametric Toolbox. We would like to note that one can use e.g. POP toolbox in the context of PAROC (Oberdieck et al., 2016) as an alternative. The corresponding explicit solution (20) was yielded in 276 s and consisted of 2979 polyhedral regions. To export such controller, one would need to store 254,304 floating points, what translates into 2034 kb of memory. This represents 2.9-times greater memory requirements compared to the region-less explicit MPC approach.

Finally, we have investigated the worst-case runtime of all three methods measured as the runtime of Algorithm 1 for region-less solutions, and as the time required to perform the sequential search through all regions in the region-based scenario. The worst-case runtimes amounted to 1.75 ms for the region-less approach implemented via Algorithm 1, 1.59 ms for the region-less technique with a complete parameterization of the primal optimizer as in (27), cf. Remark 1, and 1.16 ms for the region-based approach. The difference in the worst-case evaluation time between both region-less approaches is due to different computation of the primal optimizer $\mathbf{w}^*$ in the Step 5 of Algorithm 1, cf. Remark 1. All results are summarized in Table 1.

The memory footprint of all explicit MPC solutions can be furthermore reduced by converting double-precision (64 bit) numbers either to single- (32 bit) or to half-precision (16 bit) accuracy. Doing so on the one hand immediately reduces the memory consumption by factors of 2 and 4, respectively. On the other hand, such a reduction also inevitably introduces suboptimality. Table 2 reports achievable reduction of the memory footprint using various arithmetic precisions. Moreover, it also shows induced loss of optimality, evaluated by performing a series of closed-loop simulations for 10^4 equidistantly placed initial conditions. As can be observed, opting for single-precision arithmetics immediately reduces the memory consumption of explicit MPC solutions by a factor of 2 at virtually no loss of performance. Going for half-precision numbers brings the memory consumption down even more at the expense of only minor suboptimality. The choice of the number format is therefore an another design decision control engineers can make when aiming at implementing explicit MPC controllers on platforms with limited storage capacities.

6.2. Experimental results of closed-loop control

This section describes experimental data obtained by both PI and region-less explicit MPC policy. The objective of these controllers was to manipulate voltage of the alkaline pump p_{B} such that pH of the mixed outlet would track the desired reference. The flow rate of the acidic pump p_{A} was maintained constant with voltage $u_{\mathrm{A}} = 2.5$ V. Both experiments were initialized from the steady state $\bar{y}^s = 7$. To illustrate control performance of both control schemes, five successive reference changes were carried out, each within the interval of pH $= \bar{y}^s \pm 1$ and with time period 600 s.

The experimental data of the conventional PI closed-loop setup (see Fig. 5) with the PI controller (14) are depicted in Fig. 8. Specifically, Fig. 8a illustrates how the controlled variable $\bar{y}_{\mathrm{PI}}$ (blue-solid line) tracks the desired reference $\bar{r}$ (red-solid line), with respect to the soft constraints (black-dashed line). Furthermore, Fig. 8b shows corresponding control voltage of the alkaline pump $\bar{u}_{\mathrm{PI}}$ (cyan-solid line) with their hard constraints (black-solid line). Based on the control performance, one can observe that the settling time has decreased, compared to the open-loop response from Fig. 3. On the other hand, the main shortcomings of the proposed PI control are overshoots of the controlled variable $\bar{y}_{\mathrm{PI}}$. For example, the maximal overshoot was 20% at time 3066 s.

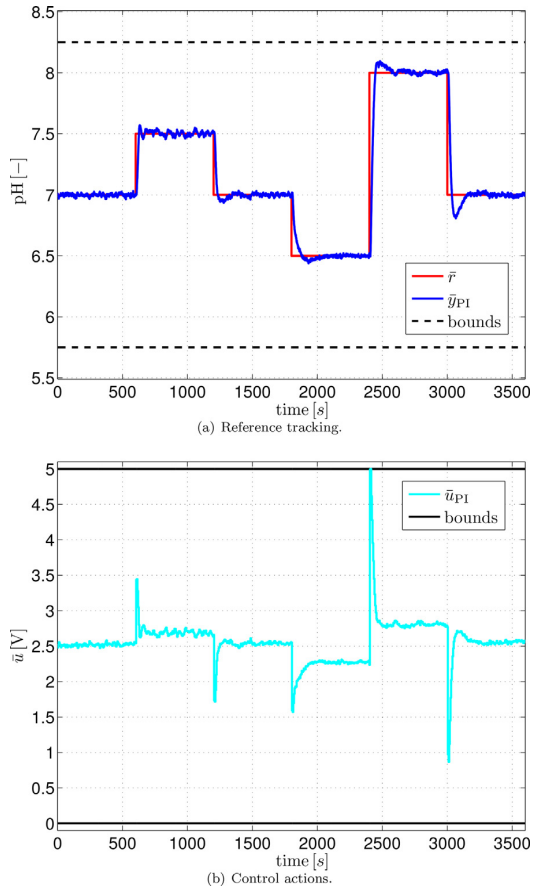
The experimental data of the reference governor paradigm (see Fig. 7), collected with the region-less explicit MPC (18), are shown

Table 1
Complexity comparison of region-less and region-based explicit MPC approaches.

Explicit MPC method	Construction time [s]	Memory footprint [kB]	Evaluation time (worst case) [ms]
Region-less via Alg. 1	208	675	1.75
Region-less via Rem. 1	209	1505	1.59
Region-based	276	2034	1.16

Table 2
Complexity comparison of region-less and region-based explicit MPC approaches w.r.t. different arithmetic precisions.

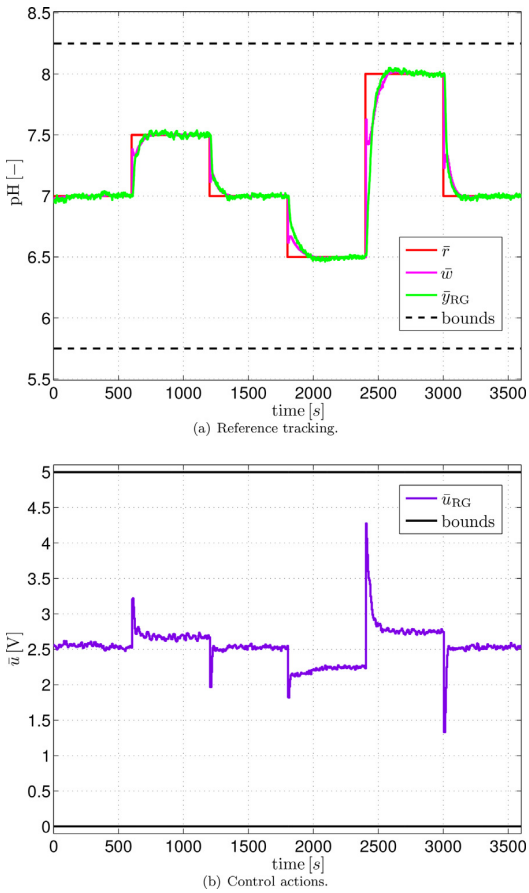
	Memory requirements [kB]			Suboptimality
Precision	Region-less via Alg. 1	Region-less via Rem. 1	Region-based	
64-bit	675	1505	2034	–
32-bit	351	753	1017	0.00%
16-bit	189	376	509	0.09%



(a) Reference tracking.

(b) Control actions.

Fig. 8. PI control of pH in the chemical vessel. (For interpretation of the references to color in text, the reader is referred to the web version of the article.)

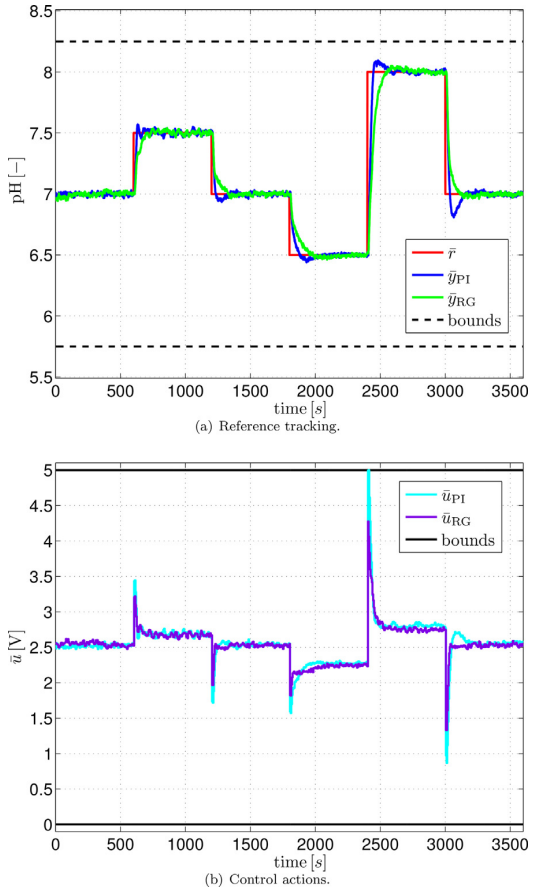


(a) Reference tracking.

(b) Control actions.

Fig. 9. Control of pH via reference governor.

in Fig. 9. Here, outputs $\bar{y}_{\mathrm{RG}}$, references $\bar{r}$, shaped references $\bar{w}$ and control inputs $\bar{u}_{\mathrm{RG}}$ are denoted by green, red, magenta and purple solid lines, respectively. Moreover, notation of soft and hard constraints is maintained the same. In figures, one can notice that profile of the controlled variable has improved in terms of suppressed overshoots. Concretely, the maximal overshoot at time 3066 s was reduced to only 4%. Yet, the settling time remained the same. The violation of the soft constraints occurred due to uncaptured nonlinear dynamics of the system and measured noise. This particular overshoot can be however easily eliminated by increasing q_s, i.e., by a greater penalization of (18i). On the other hand, as a consequence of this modification would be increased settling time in the neighborhood of the soft constraints. Next, we have used integrated squared error (ISE) criterion to characterize the consumption of the control medium (solution of sodium hydroxide). Based on this quality indicator, the reference governor scheme has decreased medium and energy consumption by almost 2%, what is mainly due to the mitigated overshoots of the controlled variable. For convenience, control performance of both approaches is shown in Fig. 10.



(a) Reference tracking.

(b) Control actions.

Fig. 10. Comparison of PI and reference governor control performance.

7. Conclusions

This paper addressed a twofold issue. The first problem elaborated the control of pH in the chemical vessel, where we aimed to improve control performance of a conventional PI control strategy. This task was achieved via employing reference governor scheme to the inner closed loop, which consisted of the controlled plant and PI controller. The objective of this additional control layer was to shape optimal references in order to supervise the PI controller, thus to enhance its performance, based on predictions of the future evolution of the controlled system. To design the prediction model for such MPC-based reference governor scheme we have firstly identified the systems dynamics via several step responses and then constructed inner loop based on the employed PI controller. Furthermore, disturbance modeling was considered to handle the process-model mismatch.

The second problem, elaborated in this paper, was dedicated to the implementation requirement reduction of the proposed reference governor scheme. In order to achieve this goal, hence to provide a fast and memory efficient control policy, we have employed the technique called region-less explicit MPC. In this method, the entire feedback law is computed offline with the use of the direct active set enumeration method, which allows one to construct an explicit solution even for large parametric space problems. Moreover, unlike in the standard region-based explicit approach, construction of polyhedral critical regions and PWA feedback law is omitted and replaced by storing only the set of all optimal active set combinations, pre-factorized dual variables of the KKT system, and matrices of the original optimization problem. This allows us to significantly decrease the memory footprint of an explicit controller. Subsequently, with all pre-computed data in hand, evaluation of optimal control actions can be then performed by means of a sequential search technique. As reported in Algorithm 1, this online evaluation relies on verifying primal and dual feasibility condition of the KKT system, what requires only a simple linear algebra. Therefore implementation of such MPC approach is fast, simple, and last, but not least, does not require additional optimization solvers.

The designed region-less MPC-based reference governor was used to control pH in the CSTR. Based on the experimental data we have shown that the control performance, compared to conventional PI closed-loop, has improved. Specifically, overshoots of the controlled variable were mitigated, what led to savings of the manipulated stream and energy, respectively. From the implementation point of view, the region-less approach has exhibited significant memory reduction. It has decreased the memory requirements by a factor of 3 when compared with the standard region-based approach. Needlessly to say, this reduction was obtained without introducing any sub-optimality of the controller. The construction time of this feedback law was almost 25% faster. On the other hand, the cost of the aforementioned decreased implementation requirements is the online evaluation time, which has increased by 50%. This, however, did not represent an issue, as the worst-case computation time of the optimal control actions was negligible compared to the systems dynamics.

Acknowledgements

The authors gratefully acknowledge the contribution of the Slovak Research and Development Agency under the project APVV-15-0007, and the financial support of the of the Scientific Grant Agency of the Slovak Republic under the grants 1/0403/15.

References

Åström, K.J., Hägglund, T., 2006. Advanced PID Control. ISA – The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC.

Altnten, A., 2007. Generalized predictive control applied to a pH neutralization process. Comput. Chem. Eng. 31 (10), 1199–1204, http://dx.doi.org/10.1016/j.compchemeng.2006.10.005 http://www.sciencedirect.com/science/article/pii/S0098135406002572.

Baotić, M., Borrelli, F., Bemporad, A., Morari, M., 2008. Efficient on-line computation of constrained optimal control. SIAM J. Control Optim. 47 (5), 2470–2489, http://dx.doi.org/10.1137/060659314.

Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N., 2002. The explicit linear quadratic regulator for constrained systems. Automatica 38 (1), 3–20.

Bemporad, A., 1998. Reference governor for constrained nonlinear systems. IEEE Trans. Autom. Control 43 (3), 415–419, http://dx.doi.org/10.1109/9.661611.

Borrelli, F., Morari, M., 2007. Offset free model predictive control. In: Proc IEEE Conference on Decision and Control., pp. 1245–1250.

Borrelli, F., Falcone, P., Pekar, J., Stewart, G., 2009. Reference governor for constrained piecewise affine systems. J. Process Control 19 (8), 1229–1237.

Borrelli, F., Baotić, M., Pekar, J., Stewart, G., 2010. On the computation of linear model predictive control laws. Automatica 46 (6), 1035–1041, http://dx.doi.org/10.1016/j.automatica.2010.02.031 http://www.sciencedirect.com/science/article/pii/S0005109810001032.

Borrelli, F., 2003. Constrained Optimal Control of Linear and Hybrid Systems, vol. 290 of Lecture Notes in Control and Information Sciences. Springer.

Chen, J., Peng, Y., Han, W., Guo, M., 2011. Adaptive fuzzy sliding mode control in pH neutralization process. Proc. Eng. 15, 954–958, http://dx.doi.org/10.1016/j.proeng.2011.08.176, CEIS, http://www.sciencedirect.com/science/article/pii/S1877705811016778.

Desborough, L., Miller, R., 2002. Increasing customer value of industrial control performance monitoring – Honeywell's experience. In: AIChE Symposium Series., pp. 169–189.

Drgoňa, J., Klaučo, M., Janeček, F., Kvasnica, M., 2017. Optimal control of a laboratory binary distillation column via regionless explicit MPC. Comput. Chem. Eng., 139–148, http://dx.doi.org/10.1016/j.compchemeng.2016.10.003 http://www.kirp.chtf.stuba.sk/assets/publication_info.php?id_pub=1768.

Fuente, M., Robles, C., Casado, O., Syafiie, S., Tadeo, F., 2006. Fuzzy control of a neutralization process. Eng. Appl. Artif. Intell. 19 (8), 905–914, http://dx.doi.

org/10.1016/j.engappai.2006.01.008 http://www.sciencedirect.com/science/article/pii/S0952197606000431.
Gal, T., Nedoma, J., 1972. Multiparametric linear programming. Manag. Sci. 18, 406–442.
Georgiev, N.I., Bryaskova, R., Tzoneva, R., Ugrinova, I., Detrembleur, C., Miloshev, S., Asiri, A.M., Qusti, A.H., Bojinov, V.B., 2013. A novel pH sensitive water soluble fluorescent nanomicellar sensor for potential biomedical applications. Bioorgan. Med. Chem. 21 (21), 6292–6302, http://dx.doi.org/10.1016/j.bmc.2013.08.064 http://www.sciencedirect.com/science/article/pii/S0968089613007670.
Gupta, A., Bhartiya, S., Nataraj, P.S.V., 2011. A novel approach to multiparametric quadratic programming. Automatica 47 (9), 2112–2117, http://dx.doi.org/10.1016/j.automatica.2011.06.019.
Gurobi Optimization, I., 2014. Gurobi Optimizer Reference Manual.
Herceg, M., Kvasnica, M., Jones, C., Morari, M., 2013. Multi-parametric toolbox 3.0. In: Proc. of the European Control Conference, Zurich, Switzerland, pp. 502–510 http://control.ee.ethz.ch/~mpt.
Hermansson, A., Syafiie, S., 2015. Model predictive control of pH neutralization processes: a review. Control Eng. Pract. 45, 98–109, http://dx.doi.org/10.1016/j.conengprac.2015.09.005 http://www.sciencedirect.com/science/article/pii/S0967066115300162.
Holaza, J., 2016. Fast and Memory-Efficient Implementation of Model Predictive Control (Ph.D. thesis), UIAM FCHPT STU v Bratislave Radlinského 9, 812 37 Bratislava (24.08.2016). http://www.kirp.chtf.stuba.sk/publication_info.php?id_pub=1754.
Ibrahim, R., 2008. Practical Modelling and Control Implementation Studies on a pH Neutralization Process Pilot Plant (Ph.D. thesis). University of Glasgow.
ILOG, Inc, 2003. CPLEX 8.0 User Manual. ILOG, Inc., Gentilly Cedex, France.
King, M., 2010. Process Control: A Practical Approach. Wiley http://books.google.sk/books?id=YKud-kwsA-4C.
Klaučo, M., Kvasnica, M., 2017. Control of a boiler-turbine unit using MPC-based reference governors. Appl. Therm. Eng. 110, 1437–1447, http://dx.doi.org/10.1016/j.applthermaleng.2016.09.041 http://www.kirp.chtf.stuba.sk/publication_info.php?id_pub=1758.
Krogell, J., Ernen, K., Pranovich, A., Willfr, S., 2015. In-line high-temperature pH control during hot-water extraction of wood. Ind. Crops Prod. 67, 114–120, http://dx.doi.org/10.1016/j.indcrop.2015.01.026 http://www.sciencedirect.com/science/article/pii/S092666901500028X.
Kvasnica, M., Takacs, B., Holaza, J., Di Cairano, S., 2015. On region-free explicit model predictive control. In: 54th IEEE Conference on Decision and Control, vol. 54, Osaka, Japan, pp. 3669–3674 http://www.kirp.chtf.stuba.sk/publication_info.php?id_pub=1675.
Mandal, S., Mahapatra, S., Patel, R., 2015. Neuro fuzzy approach for arsenic(III) and chromium(VI) removal from water. J. Water Process Eng. 5, http://dx.doi.org/10.1016/j.jwpe.2015.01.002.
Maciejowski, J., 2001. Predictive Control with Constraints. Prentice-Hall.
Muske, K.R., Badgwell, T.A., 2002. Disturbance modeling for offset-free linear model predictive control. J. Process Control 12 (5), 617–632, http://dx.doi.org/10.1016/S0959-1524(01)00051-8 http://www.sciencedirect.com/science/article/pii/S0959152401000518.
Narayanan, N.L., Krishnaswamy, P., Rangaiah, G., 1997. An adaptive internal model control strategy for pH neutralization. Chem. Eng. Sci. 52 (18), 3067–3074, http://dx.doi.org/10.1016/S0009-2509(97)00130-9 http://www.sciencedirect.com/science/article/pii/S0009250997001309.
Oberdieck, R., Diangelakis, N.A., Papathanasiou, M.M., Nascu, I., Pistikopoulos, E.N., 2016. POP – parametric optimization toolbox. Ind. Eng. Chem. Res. 55 (33), 8979–8991.
Oravec, J., Bakošová, M., 2012. PIDDESIGN – Software for PID Control Education. In: IFAC Conference on Advances in PID Control, vol. 2, Brescia, Italy, pp. 691–696.
Pannocchia, G., Rawlings, J.B., 2003. Disturbance models for offset-free model predictive control. AIChE J. 49 (2), 426–437.
Pawlowski, A., Fernndez, I., Guzmn, J., Berenguel, M., Acin, F., Normey-Rico, J., 2014. Event-based predictive control of pH in tubular photobioreactors. Comput. Chem. Eng. 65, 28–39, http://dx.doi.org/10.1016/j.compchemeng.2014.03.001 http://www.sciencedirect.com/science/article/pii/S0098135414000684.
Pistikopoulos, E.N., 2012. From multi-parametric programming theory to MPC-on-a-chip multi-scale systems applications. Comput. Chem. Eng. 47, 57–66, http://dx.doi.org/10.1016/j.compchemeng.2012.06.031 http://www.sciencedirect.com/science/article/pii/S0098135412002153.
Qian, F., Dixon, D.R., Newcombe, G., Ho, L., Dreyfus, J., Scales, P.J., 2014. The effect of pH on the release of metabolites by cyanobacteria in conventional water treatment processes. Harmful Algae 39, 253–258, http://dx.doi.org/10.1016/j.hal.2014.08.006 http://www.sciencedirect.com/science/article/pii/S1568988314001589.
Qin, S.J., Badgewell, T.A., 2003. A Survey of Industrial Model Predictive Control Technology, vol. 11., pp. 733–764.
Qin, J.-J., Oo, M.H., Kekre, K.A., Knops, F., Miller, P., 2006. Impact of coagulation pH on enhanced removal of natural organic matter in treatment of reservoir water. Sep. Purif. Technol. 49 (3), 295–298, http://dx.doi.org/10.1016/j.seppur.2005.09.016 http://www.sciencedirect.com/science/article/pii/S1383586605003187.
Salehi, S., Shahrokhi, M., Nejati, A., 2009. Adaptive nonlinear control of pH neutralization processes using fuzzy approximators. Control Eng. Pract. 17 (11), 1329–1337, http://dx.doi.org/10.1016/j.conengprac.2009.06.007 http://www.sciencedirect.com/science/article/pii/S0967066109001191.
Spjøtvold, J., Tøndel, P., Johansen, T.A., 2005. A Method for Obtaining Continuous Solutions to Multiparametric Linear Programs, Prague, Czech Republic.
The MathWorks, I., 2016. R2016a Documentation – Control System Toolbox – pidtune. http://www.mathworks.com/help/control/ref/pidtune.html.

Engineering Applications of Artificial Intelligence 77 (2019) 1–8

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Machine learning-based warm starting of active set methods in embedded model predictive control

Martin Klaučo *, Martin Kalúz, Michal Kvasnica

Slovak University of Technology in Bratislava, Radlinského 9, SK-812 37 Bratislava, Slovak Republic

ARTICLE INFO

Keywords:
Artificial intelligence
Active set method
Model predictive control

ABSTRACT

We propose to apply artificial intelligence approaches in a warm-starting procedure to accelerate active set methods that are used to solve strictly convex quadratic programs in the context of embedded model predictive control (MPC). The proposed warm-starting is based on machine learning where a good initialization of the active set method is learned from training data. Two approaches to generate the training data set are discussed, one based on gridding the feasibility domain, and one based on closed-loop simulations with typical initial conditions. The training data are then processed by machine learning-based classification algorithms that yield a good estimate of the initial active set for the iterative active set algorithm. By means of extensive case studies we demonstrate that the proposed approach is superior to existing warm-starting procedures in that it considerably reduces the number of active set iterations, thus allowing embedded MPC to be implemented using less computational effort.

1. Introduction

Model Predictive Control (MPC) is nowadays a de facto standard control methodology when controlling multivariable systems where satisfaction of constraints and optimization of the control performance are required (Maciejowski, 2002; Rossiter, 2003; Camacho and Bordons, 2007). Most of MPC in practice is formulated and solved as convex quadratic programs (QP), parameterized in the initial state measurement. When controlling systems with fast dynamics, it is vital to be able to solve such QPs fast enough. This, however, becomes challenging when constraints of the implementation hardware, such as the available computational power and memory storage, are taken into account, especially when targeting simple hardware such as microcontrollers (Zometa et al., 2013), field programmable gate arrays (Ling et al., 2008), or programmable logic controllers (Huyck et al., 2012).

Various avenues to providing the solution to a given MPC-based QP with minimal resources (computation and memory) can be pursued. One option is to solve the QP *off-line* for all possible values of the initial condition using parametric programming (Bemporad et al., 2002). Such techniques, however, are limited to problems of small size, say, for a short prediction horizon, up to 10 states and up to 4 control inputs. For problems of bigger size, one usually solves the QP *on-line*. To do so, a plethora of methods have been suggested, such as active set methods (Nocedal and Wright, 2006; Wills and Ninness, 2010; Ferreau et al., 2008), interior point approaches (Rao et al., 1998), fast gradient procedures (Richter et al., 2012; Kögel and Findeisen, 2011), gradient projection algorithms (Patrinos and Bemporad, 2014), splitting methods (O'Donoghue et al., 2013; Stathopoulos et al., 2014), and tailored MPC algorithms (Liu and Kong, 2014), to name just a few. Among these, the active set methods are frequently used due to their simplicity and easiness of implementation in the embedded framework (Cimini and Bemporad, 2017).

In this paper, we focus on active set methods (ASM). There, the working active set $\mathcal{A}$ and the primal/dual optimizers are iteratively updated until the global optimum of the convex QP is found. The dominant speed factors of ASM are the number of iterations and the cost of linear algebra in each iteration, with the former having larger impact on the overall performance (Herceg et al., 2015).

As shown in Herceg et al. (2015), the number of ASM iterations can be significantly decreased if the iterations are *warm started* from some known initial active set $\mathcal{A}_0$ and the associated primal/dual feasible solution. In Ferreau et al. (2008, 2014), the ASM is initialized from the active set $\mathcal{A}^\star(t-1)$ obtained at the previous time instant $t-1$. This, however, does not take into account the current state measurement and therefore works well only when the active sets do not change much in time. An improved warm starting procedure, which takes into account the information about the current state $x(t)$, was proposed in

* Corresponding author.
E-mail addresses: martin.klauco@stuba.sk (M. Klaučo), martin.kaluz@stuba.sk (M. Kalúz), michal.kvasnica@stuba.sk (M. Kvasnica).

https://doi.org/10.1016/j.engappai.2018.09.014
Received 2 March 2018; Received in revised form 13 July 2018; Accepted 18 September 2018

Otta et al. (2015). There, LQR-based warm starting is used in conjunction with projections onto constraints. Therefore the approach is only suitable, from a practical point of view, if the constraint set is simple, such as a hyperbox. Finally, in Zeilinger et al. (2011) the authors have proposed to devise a state-dependent warm start $\mathcal{A}_0(x(t))$ by solving, off-line, a parametric program. Although substantial reduction in the number of iterations could be achieved, the procedure is limited to systems with a modest number of states, say, below ten.

In this paper we propose to apply machine learning (ML) to accelerate primal active set methods used to solve QP-based MPC problems. Specifically, we show how to construct a state-dependent warm start procedure that yields the initial active set $\mathcal{A}_0$ as a function of the current state $x(t)$. Specifically, the policy $\mathcal{A}_0(x(t))$ is learned from training data collected off-line. We show that by resorting to standard ML-based classification algorithms, such as classification trees (Breiman et al., 1984) and k-nearest neighbors (Dasarathy, 1991), the learned policy is simple enough as to enable a fast and cheap embedded implementation, and performs better compared to the feedback warm-starting of Ferreau et al. (2008). Finally, the procedure is applicable to QP-based MPC problems where the constraint set is an arbitrary polyhedron, thus allowing, among others, to include terminal set constraints. In fact, application of machine learning procedures to solve optimization problems is not new. Research presented in Nazemi (2014) and Elsayed et al. (2014) use the neural networks to find optimal solution to the strictly convex QPs.

Note that the proposed machine-learning based acceleration of active set methods *does not replace* the control algorithm, it merely complements it. Specifically, as shown in Section 5, ML allows the ASM to arrive at the optimal solution using a fewer number of iterations compared to conventional methods, such as the one of Ferreau et al. (2008). Since the primal active set method guarantees that the optimal solution is found even if the initialization is not correct, the proposed procedure maintains guarantees of closed-loop stability and constraint satisfaction.

2. Preliminaries

2.1. QP-based MPC

For linear systems

$$x(t+1) = Ax(t) + Bu(t), \tag{1}$$

subject to constraints $x \in \mathcal{X}$, $u \in \mathcal{U}$ with $\mathcal{X} \subseteq \mathbb{R}^n$, $\mathcal{U} \subseteq \mathbb{R}^m$, the context of this paper revolves around solving the following MPC problem:

$$\min_{u_0,\dots,u_{N-1}} \; x_N^\intercal Q_\mathrm{f} x_N + \sum_{k=0}^{N-1} \left(x_k^\intercal Q_\mathrm{x} x_k + u_k^\intercal Q_\mathrm{u} u_k\right) \tag{2a}$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k, \quad k = 0,\dots,N-1, \tag{2b}$$

$$(x_k, u_k) \in \mathcal{X} \times \mathcal{U}, \qquad k = 0,\dots,N-1, \tag{2c}$$

$$x_N \in \mathcal{X}_\mathrm{f}, \tag{2d}$$

where $Q_\mathrm{f} \succeq 0$, $Q_\mathrm{x} \succeq 0$, $Q_\mathrm{u} \succ 0$ are penalty matrices, $N \in \mathbb{N} < \infty$ is a finite prediction horizon, A, B are system matrices, and $\mathcal{X}$, $\mathcal{U}$, $\mathcal{X}_\mathrm{f}$ are polyhedra that contain the origin in their respective interiors, and $\mathcal{X}_\mathrm{f} \subset \mathcal{X}$. It is well known, see, e.g., Mayne et al. (2000) that (2) can be converted into a parametric quadratic program

$$\min_U \; 1/2\, U^\intercal H U + \theta^\intercal F U \tag{3a}$$

$$\text{s.t. } GU \leq E\theta + w, \tag{3b}$$

with $\theta = x_0$ being the parameter and $U = [u_0^\intercal, \dots, u_{N-1}^\intercal]^\intercal$ representing the optimization variables. The vectors/matrices H, F, G, E, w can be obtained from (2) using straightforward algebraic manipulations. We remark that $H \succ 0$, thus the Hessian is invertible, since $Q_\mathrm{f} \succeq 0$, $Q_\mathrm{x} \succeq 0$, $Q_\mathrm{u} \succ 0$ is assumed. The optimal open-loop sequence of control actions, i.e., $U^\star$, can then be obtained by solving (3) for a particular initial condition $\theta = x(t)$. MPC is then traditionally implemented in the receding horizon fashion where only the first element of $U^\star$, i.e., $u_0^\star$ is implemented to the controlled system and the whole procedure is repeated at the subsequent time instant for a new value of the initial condition $\theta = x(t)$.

2.2. Primal active set method for strictly convex QPs

The primal active set method (ASM) (Fletcher, 2013) is an iterative procedure for finding the minimizer $U^\star$ to (3) for a fixed value of the parameter θ. At each iteration, the ASM searches for a vector Δ along which the objective function (3a) decreases. To tackle constraints, the procedure also iteratively updates the index set $\mathcal{A}$ of constraints that are active at the current iterate. Specifically, at each iteration the active set method solves the equality-constrained QP (EQP) of the form

$$\min_\Delta \; \frac{1}{2}(U+\Delta)^\intercal H (U+\Delta) + \theta^\intercal F (U+\Delta) \tag{4a}$$

$$\text{s.t. } G_\mathcal{A}(U+\Delta) = E_\mathcal{A}\theta + w_\mathcal{A}, \tag{4b}$$

where $G_\mathcal{A}$ consists of the rows of G indexed by the set $\mathcal{A} \subseteq \{1,\dots,c\}$ where c is the number of constraints in (3b). In (4), U is considered fixed and feasible, i.e., $G_\mathcal{A} U = E_\mathcal{A}\theta + w_\mathcal{A}$, and therefore $G_\mathcal{A}\Delta = 0$ needs to hold to retain feasibility. The improving direction Δ for (4) can be solved from the Karush–Kuhn–Tucker system

$$\begin{bmatrix} \Delta \\ \lambda \end{bmatrix} = \begin{bmatrix} H & G_\mathcal{A}^\intercal \\ G_\mathcal{A} & 0 \end{bmatrix}^{-1} \begin{bmatrix} -HU - F^\intercal\theta \\ 0 \end{bmatrix}, \tag{5}$$

where λ is the vector of Lagrange multipliers. If[1] $\|\Delta\| = 0$ and $\lambda_i \geq 0$ for all $i \in \mathcal{A}$, U is the optimal solution and the iterations terminate. If, on the other hand, some Lagrange multipliers are negative, one constraint is removed from $\mathcal{A}$, typically the one corresponding to the smallest Lagrange multiplier, i.e., $\mathcal{A} = \mathcal{A} \setminus \{i^\star\}$ with

$$i^\star = \arg\min_{i \in \mathcal{A}} \lambda_i, \tag{6}$$

and the procedure continues with the next iteration. If the improving direction Δ is non-zero, the current iterate U is refined by $U = U + \alpha\Delta$. Here, the step size α is given by $\alpha = \min\{1, \beta_j\}$, where β_j are defined, for all $j \notin \mathcal{A}$ for which $G_j\Delta > 0$, as

$$\beta_j = \frac{E_j\theta + w_j - G_j U}{G_j \Delta}, \tag{7}$$

where G_j is the jth row of the matrix. If $\alpha < 1$, then some previously inactive constraint becomes active when updating U along the direction Δ. Among all such constraints, one typically (Fletcher, 2013) picks the one that is activated first, i.e., $j^\star = \arg\min_j \beta_j$, followed by adding $j^\star$ to the active set, i.e., $\mathcal{A} = \mathcal{A} \cup \{j^\star\}$. The complete procedure is reported as Algorithm 1.

Remark 2.1. Algorithm 1 takes the initial active set $\mathcal{A}_0$ and the initial primal solution U_0 as its inputs. If no prior information is available, one can always choose $\mathcal{A}_0 = \emptyset$ and $U_0 = 0$, provided $0 \in \mathcal{U}$. If $\mathcal{A}_0$ is given as a non-empty set, U_0 can be computed by $U_0 = G_{\mathcal{A}_0}^\dagger (E_{\mathcal{A}_0}\theta + w_{\mathcal{A}_0})$ where $\dagger$ denotes the left Moore–Penrose inverse. □

Remark 2.2. As pointed out in Fletcher (2013, Chapter 10.3), Algorithm 1 converges to the primal optimal solution $U^\star$ regardless of the choice of the initial active set $\mathcal{A}_0$, provided it is chosen such that the matrix $G_{\mathcal{A}_0}$ is of full row rank. Naturally, each choice of $\mathcal{A}_0$ leads to a different sequence of iterations. □

3. Problem statement

The number of iterations the ASM takes to converge to the optimal solution $U^\star(t)$ for a given initial condition $\theta = x(t)$ at time instant t

[1] In the floating point environment, it is advised to replace the condition $\|\Delta\| = 0$ in Algorithm 1 by $\|\Delta\| \leq \epsilon$ for some small positive value of ϵ, typically equal to the machine precision.

Algorithm 1 Active-Set Method

```
Choose initial active set A_0 and a feasible solution U_0
Set A ← A_0, U ← U_0
while not converged do
    Solve Eq. (5) for Δ and λ
    if ‖Δ‖ = 0 then
        if λ ≥ 0 then
            return U* = U
        else
            A ← A \ {i*} per Eq. (6)
        end if
    else
        Determine β_j per Eq. (7) for j ∉ A, G_j Δ > 0
        Set α = min{1, β_j}
        Update U ← U + αΔ
        if α < 1 then
            A ← A ∪ {j*} where j* = arg min_j β_j
        end if
    end if
end while
```

depends on the choice of the initial active set $\mathcal{A}_0 \subseteq \{1,\dots,c\}$, where c is the number of constraints in (3b). With no a-priori knowledge, $\mathcal{A}_0$ is typically chosen as an empty set $\mathcal{A}_0 = \emptyset$, along with $U_0 = [0,\dots,0]^\intercal$ as the initial feasible solution. Such a choice is referred to as *cold-starting*.

Alternatively, the ASM can be *warm-started* from some non-empty initial active set $\mathcal{A}_0 \neq \emptyset$ using prior knowledge. In the context of feedback implementation of MPC, given $U^\star(t-1)$ and $\mathcal{A}^\star(t-1)$ as respective quantities obtained at the previous time instant $t-1$ for $\theta = x(t-1)$, $U^\star(t)$ can be computed by invoking Algorithm 1 with $\mathcal{A}_0 = \mathcal{A}^\star(t-1)$ and $U_0 = U^\star(t-1)$, see Ferreau et al. (2008). There is, however, no guarantee that such a feedback-based warm-start is better than cold-starting, especially when the active set $\mathcal{A}^\star(t)$ differs significantly in cardinality from $\mathcal{A}^\star(t-1)$ (e.g., when $\theta = x(t)$ is far away for $\theta = x(t-1)$). In fact, it could happen that such a warm-starting *increases* the number of iterations as compared to cold-starting, as will be demonstrated in Section 5.

In this paper we propose a novel warm-starting approach based on machine learning that will reduce the number of iterations of the primal active set method. This will be achieved by constructing a procedure that yields $\mathcal{A}_0$ as a function of the parameter θ in (3). Specifically, we seek the mapping $\kappa : \mathbb{R}^n \to \{1,\dots,c\}$ such that $\mathcal{A}_0 = \kappa(\theta)$ allows the ASM implemented per Algorithm 1 to converge faster compared to cold-starting and traditionally used feedback warm-starting of Ferreau et al. (2008). Moreover, in the spirit of embedded MPC, it is desirable for $\kappa(\cdot)$ to have a low memory footprint and being fast to evaluate. In what follows we propose to construct the mapping $\kappa(\cdot)$ by using machine learning-based classification.

Remark 3.1. All procedures of this paper are directly applicable to accelerate, in terms of reducing the number of iterations, dual active set methods (Wills and Ninness, 2010), dual gradient projection methods (Axehill and Hansson, 2008), online active set strategies (Ferreau et al., 2008), as well as spline-based methods of Li and Swetits (1997). □

Remark 3.2. Unlike the warm-starting method proposed in Ferreau et al. (2008), which is tailored to control-oriented setups, the methods of this paper are also applicable to accelerate generic QP problems of the form (3) that need to be solved multiple times for various values of the parameter, say, $\theta(t-1)$ and $\theta(t)$, with $\theta(t-1)$ having *no direct relation* to $\theta(t)$. Such problems often occur, e.g., in economics (Dorfman et al., 1958) and portfolio optimization (Wu and Liu, 2012), or in analysis of elasto-plastic structures (Wanxie and Suming, 1988). □

The proposed machine learning-based warm starting of the active set method only serves to reduce the number of iterations in Algorithm 1, thus to reduce the time needed to obtain the optimal control actions $U^\star$ in the MPC problem (2). Since Algorithm 1 always converges to $U^\star$ irrespective of the choice of the initial active set (cf. Remark 2.2), the proposed warm starting can never have a negative impact on control stability. Specifically, stability of the MPC controller only depends on the choice of Q_f and $\mathcal{X}_\mathrm{f}$ in (2a) and (2d), respectively, see, e.g., Maciejowski (2002).

Moreover, satisfaction of hard limits on states and inputs, represented by the constraint sets $\mathcal{X}$ and $\mathcal{U}$ in (2c), is also preserved under the proposed method as it only depends on Algorithm 1 computing the optimal control actions $U^\star$ that satisfy (2b)–(2d). As pointed out above, this is always guaranteed by Algorithm 1 irrespective of the choice of the initial active set $\mathcal{A}_0$ by the arguments of Remark 2.2.

Robustness in terms of rejection of disturbances can be addressed by a straightforward modification of the MPC problem in (2) by the arguments of robust MPC as proposed in Mayne et al. (2000). Specifically, the controller can be made robust against bounded additive disturbances by contracting the state constraint set $\mathcal{X}$. Note that even after this modification the MPC problem in (2) remains a convex quadratic program as in (3) and can therefore be solved by Algorithm 1 accelerated by the proposed warm starting based on machine learning.

4. Machine learning-based classification

Assume we are given a set of M distinct realizations of the parameter θ in (3), i.e, $\{\theta_{(1)},\dots,\theta_{(M)}\}$, along with the set of respective[2] optimal active sets $\{\mathcal{A}^\star_{(1)},\dots,\mathcal{A}^\star_{(M)}\}$. Henceforth, we will refer to the set of tuples $\{(\theta_{(i)},\mathcal{A}^\star_{(i)})\}$ as the *training data*. In machine learning-based classification, the task is to construct the classifier $\kappa(\theta)$ that identifies to which active set a new observation θ belongs to, by minimizing the classification error among the training data, i.e.,

$$\min_{\kappa} \sum_{i=1}^{M} \|\kappa(\theta_{(i)}) - \mathcal{A}^\star_{(i)}\|. \tag{8}$$

The construction of the classifier κ is a two step procedure that is performed *off-line*. In the first step, the training data set is created. Then, in the second step, the classifier is designed by solving (8). The generation of the training data set is tackled in Section 4.1 while two standard methods for constructing the classifier are reviewed in Section 4.2.

4.1. Generation of training data

In this section we review two methods of generating the training data set $\{(\theta_{(i)},\mathcal{A}^\star_{(i)})\}$. The first method is based on selecting M distinct samples $\theta_{(i)}$ from the set Ω of parameters for which (3) is feasible, i.e.,

$$\Omega = \{\theta \mid \exists U \text{ s.t. } GU \leq E\theta + w\}. \tag{9}$$

Such a set can be constructed off-line by projecting the polyhedron $\mathcal{P} = \{(U,\theta) \mid GU - E\theta \leq w\}$ onto the θ-space using, for instance, the Fourier–Motzkin elimination (Dantzig and Eaves, 1973), or the equality set projection method (Jones et al., 2004). A Matlab-based implementation of these methods is provided in the Multi-Parametric Toolbox (Herceg et al., 2013). Once the set Ω is constructed, the samples $\theta_{(i)}$, $i = 1,\dots,M$ can be selected, e.g., by equidistantly gridding the set Ω. Subsequently, for each sample $\theta_{(i)}$, the associated optimal active set $\mathcal{A}^\star_{(i)}$ is obtained from Algorithm 1. We reiterate that such a construction is performed off-line, and therefore even large values of M can be considered.

The downside of such a procedure it that computing Ω by projection is hard. Specifically, assume that $\mathcal{P}$ is given by c inequalities as in (3b), $U \in \mathbb{R}^{Nm}$ (here, N is the prediction horizon and m is the number of inputs of the system), and $\theta \in \mathbb{R}^n$. Then eliminating $d = Nm - n$ dimensions from $\mathcal{P}$ (i.e., projecting $\mathcal{P}$ from the $U-\theta$ space onto the θ

[2] Notice that multiple distinct parameters can yield the same active set.

space) can result in at most $4(c/4)^{2^d}$ inequalities that define Ω, a double exponential complexity. Therefore the gridding-based approach is only suitable, from a practical point of view, for problems of small size, say, $n < 10$, $Nm < 30$.

Alternatively, samples $\theta_{(i)}$ can be drawn from an outer approximation of Ω that is easy to compute, such as the box approximation

$$\overline{\Omega} = \{\theta \mid \underline{\theta} \leq \theta \leq \bar{\theta}\}. \tag{10}$$

Here, $\underline{\theta} \in \mathbb{R}^n$ and $\bar{\theta} \in \mathbb{R}^n$ can be computed by

$$\underline{\theta}_j = \min_{\theta,U}\{\theta_j \mid GU \leq E\theta + w\},\ j = 1,\ldots,n, \tag{11}$$

and

$$\bar{\theta}_j = \max_{\theta,U}\{\theta_j \mid GU \leq E\theta + w\},\ j = 1,\ldots,n, \tag{12}$$

where θ_j is the jth element of the vector. Each of (11)–(12) is a linear program (LP) in decision variables θ and U. Thus the outer box approximation $\overline{\Omega}$ of (9) can be computed with a total of $2n$ LPs. Since $\overline{\Omega} \supseteq \Omega$, not all samples of θ selected from $\overline{\Omega}$ are feasible for (3). In such a case one only adopts the samples $\theta_{(i)}$ that are feasible. Testing feasibility of $\theta_{(i)}$ can be done by solving a feasibility LP of finding U s.t. $GU \leq E\theta_{(i)} + w$ holds.

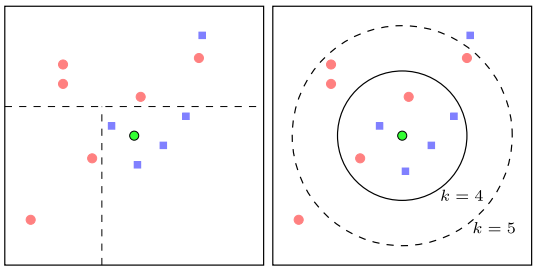
The second method is to determine the samples $\theta_{(i)}$ by performing, off-line, closed-loop simulations of the system (1) subject to the MPC feedback $u^\star(t) = u_0^\star(x_0)$ from (2). Here, we first select a set of feasible starting points $x_{(j)}(0)$ for $j = 1,\ldots,\widetilde{M}$ using knowledge of the system in (1), e.g., by considering "typical" initial conditions that are going to be likely used in practice. Then, for each $x_{(j)}(0)$ one computes the sequence of points $x_{(j)}(t+1) = Ax_{(j)}(t) + Bu^\star(t)$ for $t = 0,\ldots,T-1$ (here, T denotes the number of simulation steps) where at each step of the simulation $u^\star(t) = u_0^\star$ is obtained by solving (2) using Algorithm 1 for the initial condition $x_0 = x_{(j)}(t)$. As a consequence, one obtains a set of closed-loop state trajectories $\{x_{(j)}(0),\ldots,x_{(j)}(T)\}$ along with the associated optimal active sets $\mathcal{A}^\star_{(j)}(t)$. Finally, one takes as the training data the set of $\theta_{(i)} = x_{(j)}(t)$ and $\mathcal{A}^\star_{(i)} = \mathcal{A}^\star_{(j)}(t)$ for each $j = 1,\ldots,\widetilde{M}$ and $t = 0,\ldots,T-1$. The total number of training data points is thus $M = \widetilde{M}T$. The advantage of this method is that the sample points $\theta_{(i)}$ are representative with respect to the evolution of the controlled system (1) subject to the MPC controller (2).

4.2. Classifier construction

Given the training set $\{(\theta_{(i)}, \mathcal{A}^\star_{(i)})\}$, the function κ that classifies a new sample θ to one of the active sets can be obtained from (8). The class of classifiers that are easy to construct and yield a good performance in terms of (8) includes, among others, classification trees (Breiman et al., 1984), k-nearest neighbors (Dasarathy, 1991), naïve Bayes classifiers (Zhang, 2004), support vector machines and ensembles thereof Cristianini and Shawe-Taylor (2000), random forests (Pal, 2005), and neural networks (Gurney, 2003). Among these, only the classification tree (CT) and the k-nearest neighbors (k-NN) approaches are pursued here, as they were the most efficient ones for the types of problems considered in this paper in terms of a low classification error and a low complexity of the classifier.

4.2.1. Classification trees

Classification trees consist of a set of splitting functions $\sigma_j : \mathbb{R}^n \to \mathbb{R}$, organized as a binary tree where each leaf node represents one predicted optimal active set. In each non-leaf node, the sign of $\sigma_j(\theta)$ determines whether the next visited node is the one on the left or on the right. The splitting functions σ_j are typically restricted to linear, axis-aligned functions for simplicity of the construction. An example of a classification tree for a binary classification is shown in Fig. 1(a). The accuracy of the tree can be adjusted by *hyperparameter optimization*, during which parameters like the minimal amount of samples in each leaf node, and maximal number of splits, are selected in an optimal fashion.



(a) Classification tree. (b) k-NN classification.

Fig. 1. Visualization of two classifiers. Blue squares represent label 1 and red discs correspond to label 2. The green disc is the unlabeled sample θ to be classified. In Fig. 1(b), the dashed lines represent two splitting functions σ_1, σ_2. Fig. 1(b) shows the k-NN classifier. If $k = 4$, then the unlabeled green point will be assigned to label 1, as it has 4 blue points closer to it than the red points. If $k = 5$, then the point is classified as label 2 since it has 5 red points closer to it than the blue points. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.2.2. K-nearest neighbors classifier

In the k-NN classification an unlabeled data point θ is assigned to a corresponding active set (the label) based on the distance from θ to individual samples $\theta_{(i)}$. Specifically, for a given θ, one identifies k training data points that are closest to θ in some measure. Then, θ is classified to belong to the active set which is most frequent among the k nearest samples. A graphical illustration of the procedure is shown in Fig. 1(b). The constant k in k-NN classification is either specified by the user or, alternatively, can be determined using *hyperparameter optimization*. Moreover, the performance of the classification can be adjusted by selecting the norm in which the distance between θ and $\theta_{(i)}$ is evaluated when searching for the k nearest neighbors.

4.3. Complete procedure

The proposed ML-based warm-starting procedure consists of a learning phase and an implementation phase. The learning phase is performed off-line and first requires the generation of the training data set $\{(\theta_{(1)}, \mathcal{A}^\star_{(1)}),\ldots,(\theta_{(M)}, \mathcal{A}^\star_{(M)})\}$ per Section 4.1, followed by the construction of the classifier κ per Section 4.2.

In the implementation phase, the trained classifier assigns an unknown data point θ to one of the optimal active sets $\mathcal{A}^\star_{(1)},\ldots,\mathcal{A}^\star_{(M)}$ by evaluating $\kappa(\theta)$. In the best case, $\kappa(\theta) = \mathcal{A}^\star(\theta)$ and the on-line active set method represented by Algorithm 1 converges in one iteration when initialized by $\mathcal{A}_0 = \kappa(\theta)$. In general, however, $\kappa(\theta) \neq \mathcal{A}^\star(\theta)$. In this case, Algorithm 1 will need more than one iteration to converge to the global optimum. However, as shown in Section 5, the number of iterations is usually significantly lower compared to the warm-starting of Ferreau et al. (2008).

In the context of feedback implementation of MPC, the implementation phase consists of the following steps:

1. measure the state of the system $\theta = x(t)$;
2. evaluate the classifier $\mathcal{A}_0 = \kappa(\theta)$;
3. run Algorithm 1 initialized by $\mathcal{A}_0$, obtain $U^\star = \{u_0^\star,\ldots,u_{N-1}^\star\}$;
4. implement the control input $u(t) = u_0^\star$ to the system (1) and repeat from Step 1 at the subsequent time instant.

5. Case study

5.1. Illustrative case study for a SISO system

A double integrator system with $A = \left[\begin{smallmatrix}1.0 & 1.0\\ 0.0 & 1.0\end{smallmatrix}\right]$, $B = \left[\begin{smallmatrix}1.0\\ 0.5\end{smallmatrix}\right]$ is considered for the construction of the model predictive controller (MPC) in (2),

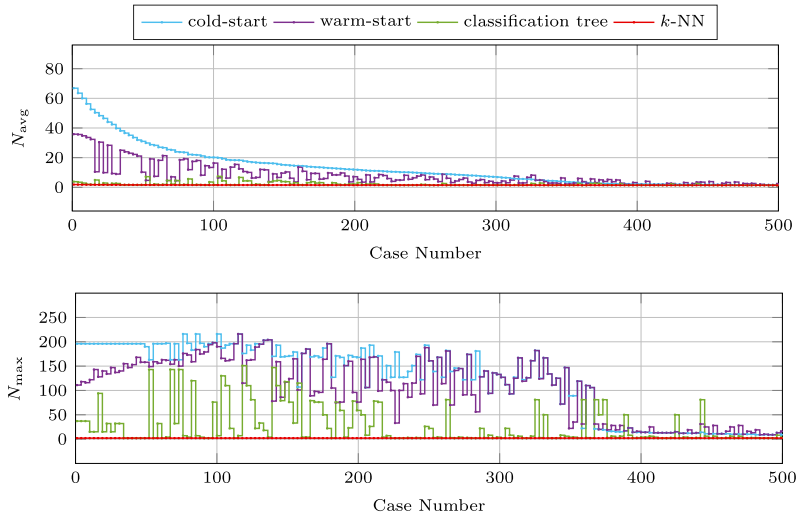


Fig. 2. Variable N_{avg} shows the average number of iterations per one sample in a closed-loop simulation, and the N_{max} is the maximum number of iterations per one closed-loop simulation. The closed-loop simulations involve the regulation of the double integrator system using MPC feedback policy.

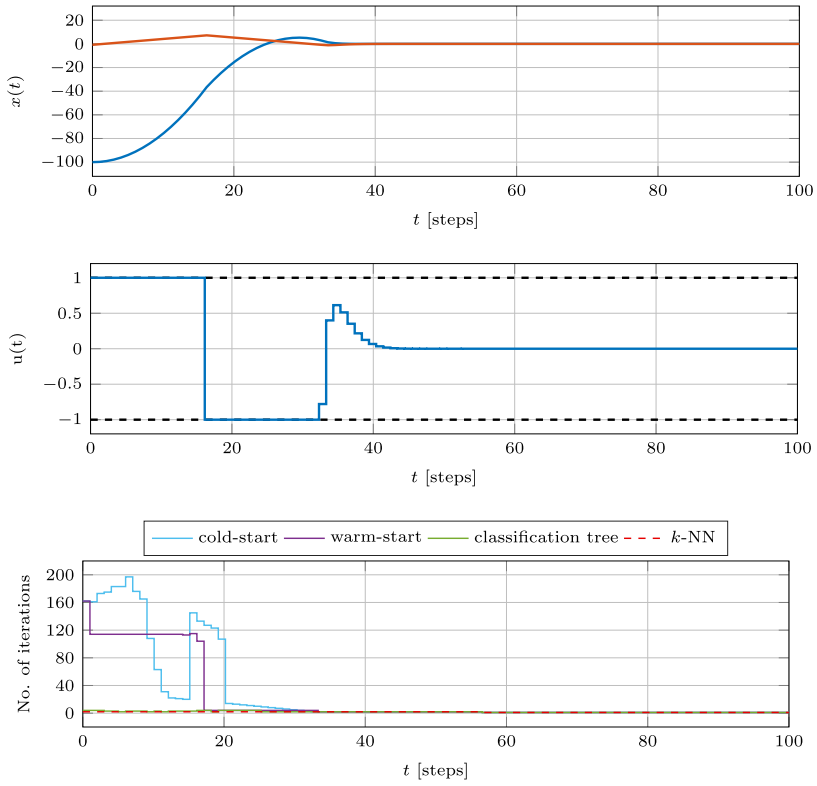


Fig. 3. Comparison of the number of iterations needed per one simulation step for the case No. 465. The blue solid line represents performance of the cold-started ASM, the pink solid line the warm-started ASM, the solid green line depicts the number of iterations of warm-started ASM via a classification tree, while the red dashed line represent warm-starting by the k-NN. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

with prediction horizon $N = 60$. Weighting matrices Q_f, Q_x, and Q_u were set to identity matrices of appropriate size. Moreover, we consider a constraint on the control action $u \in [-1, 1]$ and states $[-100, -100]^T \leq x \leq [100, 100]^T$. The optimization problem in (3) consist of 60 optimization variables and of 242 inequality constraints.

The data points in the training data set were obtained by equidistantly gridding the feasibility domain Ω (cf. Section 4.1, Eq. (9)), which was obtained via MPT3 toolbox (Herceg et al., 2013). Moreover, we have also constructed a testing data set, against which we evaluate the performance of the warm-started ASM. Specifically, by the equidistant

gridding we obtained 258 initial conditions. Subsequently, we have performed 258 closed-loop simulations. The length of each simulation was 50 samples. For each simulation step, we collected the set of active constraints, which results in 1035 unique combinations of active constraints. Hence, we obtained the training data set which consists of 13 158 data points. Next, we constructed two classifiers with hyper-parameter optimization. First, the classification tree was obtained in 19 min and consisted of 2043 nodes. Second, the k-NN classifier was constructed in 6 min, which consisted of 13 158 observation. Both classifiers were obtained using the *Statistics and Machine Learning Toolbox* in MATLAB R2017b on Intel Core i5 machine with 8 GB of memory.

The performance of the warm-started active set method (ASM) enhanced by the classification tree (CT) and the k-NN classifiers was evaluated against a test data set consisting of 500 initial conditions. This many closed-loop simulations with 100 steps were performed with the MPC posed as a QP which was solved using a cold-started ASM, a warm-started ASM as described in the Section 3, warm-started ASM with CT, and warm-started ASM with k-NN. Results reported in Fig. 2, show the cumulative number of iterations required by each of the ASM methods to converge to the optimal solution. We can observe that in all cases, the k-NN classifier provides an optimal estimate of the active set, hence it outperforms the standard warm-starting as well as the cold-started ASM. Moreover, the usage of CT in the warm-starting ASM decreases the number of iterations in 98.2% cases.

Furthermore, we can observe, that by considering standardized warm-starting procedure of the ASM, the total number of iterations is increased in 17.6% of all test cases compared to the cold-started ASM. Such a behavior is captured on the closed-loop profile of the case number No. 465. The initial condition for this closed-loop profile was set to $x(0) = [-100.0 \ -0.7]^\intercal$. The Fig. 3 shows the trajectories of states, control input and the number of iterations needed for the ASM method to converge, respectively. In general, the ML-based ASM requires less than 3 iterations to converge during the entire simulation. On the other hand, in the time instants 10 to 16 and from 30 to 34 the cold-started ASM performs better than the standardized warm-started ASM. The main reason behind this behavior, is the initialization of the ASM algorithm 1 by indexes of active constraints from the previous time instant. At the time step $t = 33$, the value of the state vector was $x(33) = [2.9173, \ -0.7]^\intercal$, and the optimal control action was $u^\star(33) = -0.7795$. The corresponding optimal active set is in this case an empty set, however the standardized ASM was initialized by $\mathcal{A}^\star(32) = \{4\}$. Hence, it required 4 iteration in total to converge. However, the cold-started ASM, as well as ASMs enhanced via machine learning converged in 2 iterations.

Although counterintuitive, the warm starting of Ferreau et al. (2008) sometimes causes Algorithm 1 to perform more iterations compared to cold starting, as can be seen in Fig. 2. Our explanation for this behavior is as follows. First, note that the warm-starting of Ferreau et al. (2008) uses the optimal active set $\mathcal{A}^\star(t-1)$ obtained at the previous time instant for $\theta(t-1)$ to initialize Algorithm 1. However, since MPC is applied in discrete time, the current state measurement $\theta(t)$ can, in fact, be far away from $\theta(t-1)$. Therefore $\mathcal{A}^\star(t)$ can be very different (in terms of cardinality, hence in terms of the number of iterations of Algorithm 1) from $\mathcal{A}^\star(t-1)$. Therefore if the cardinality of $\mathcal{A}^\star(t-1)$ is large and the cardinality of $\mathcal{A}^\star(t)$ is small, one would be better of with initializing Algorithm 1 with an empty active set (thus performing cold-starting) as it would allow Algorithm 1 to converge to $\mathcal{A}^\star(t)$ quicker. Evidently, as observed in Fig. 2, this is often the case. On the other hand, the proposed machine learning-based warm starting procedure is parameterized in the current measurements $\theta(t)$, therefore it yields better estimates of the initial active set, thus allowing Algorithm 1 to arrive at the optimal solution faster.

Note that the closed-loop trajectories obtained by solving for the optimal control actions by running Algorithm 1 initialized in different ways (cold start, warm start of Ferreau et al. (2008), as well as the proposed machine learning-based warm start) coincide. This is a natural consequence of Remark 2.2 which states that Algorithm 1 always converges to the same optimal solution regardless of the choice of the initial active set.

Table 1
Influence of the training data size on the quality of the ML-based initialization, expressed by an average number of iterations N_{avg}. Results are obtained for the simulation case No. 465. Note that the cold-started ASM and the warm-starting of Ferreau et al. (2008) require 267.2 and 213.3 iterations on average, respectively.

Number of samples	N_{avg}		Size [kB]	
	CT	k-NN	CT	k-NN
1 020	15.1	1.8	1 648	47
1 479	12.7	1.8	2 433	70
1 836	11.6	1.8	3 025	86
3 060	7.0	1.8	5 133	133
4 488	4.9	1.8	7 482	208
6 222	2.8	1.8	10 234	290
8 262	1.9	1.8	13 579	391
11 679	1.9	1.7	19 109	565
13 158	1.8	1.6	21 625	613

Naturally, the quality of the ML-based warm-starting depends on the size of the training data set. The influence of the size of the training set on the quality of the ML-based warm-start is reported in Table 1. As can be seen, the k-NN classifier performs fairly well already for a small number of training samples and provides a near-to-optimal estimates of active constraints, regardless of the size of the training set. The classification tree, on the other hand, befits greatly from more training data. Needless to say, the larger the training data set, the larger is the memory footprint of respective classifiers.

Note, that results reported in Fig. 2, Fig. 3 and Table 1 were obtained by performing simulations with MATLAB R2017b on Intel Core i5 machine with 8 GB of operational memory.

To measure the computational load of proposed enhancements to the ASM, we recoded the procedure in Section 4.3 to C code using the built-in Matlab function "codegen". A target device selected for ASM evaluation was a single-board computer based on Broadcom BCM2835 SoC (System-on-a-chip) with 700 MHz CPU (ARMv6 architecture) and 256 MB of RAM. The C code for each method was compiled directly on the target device using a GCC compiler, using compiler's optimization setting O3, which improves the source code performance. By considering the compilation option for size optimization, Os, we decreased the size by approximately 1%, which had a non-measurable effect on the execution time. Subsequently, we performed two closed-loop simulations with initial conditions $x(0) = [-100.0, \ -0.7]^\intercal$ (case No. 465), and $x(0) = [-88.6, \ 0.3]^\intercal$ (case No. 108) to measure the execution time of the optimization using 4 considered methods (cf. Fig. 3). Table 2 shows alongside the memory requirements, also the longest execution time per one time step and also the average time required by each individual method. The utilization of machine learning enhancements of the ASM method decreases the worst time case by a factor of 20, while on average by a factor of 3.

5.2. Illustrative case study for a MIMO system

A larger case study involves a model of a Boeing 747 aircraft. The states vector of this model consist of the airspeed in horizontal direction, the airspeed in vertical direction, the rotation of the aircraft, and the horizontal nose-tip angle. Manipulated variables are elevator angle and the thrust. Finally, the measured variables are true air speed and the climb rate. Further details about this benchmark model can be found in Camacho and Bordons (2007). For illustration purposes, we introduce a discretized state space model, using the sampling time of 0.2 s, expressed

Table 2
Computational load comparison of the embedded implementation of procedure in the Section 4.3 on Broadcom BCM2835 chip (700 MHz CPU). For classification-tree application two cases are tested (cf. Fig. 2). Case no. 465 represents the situation where the CT approach outperforms the k-NN classifier. Case no. 108 represents a worst case for the CT classifier.

Type of ASM	Size [kB]	Execution time [ms]			
		Worst case		Average time	
		No. 465	No. 108	No. 465	No. 108
Cold-start	171	4032	4361	453	545
Warm-start of Ferreau et al. (2008)	176	3325	3005	375	429
CT warm-start	21 625	167	1002	117	257
k-NN warm-start	613	168	169	144	144

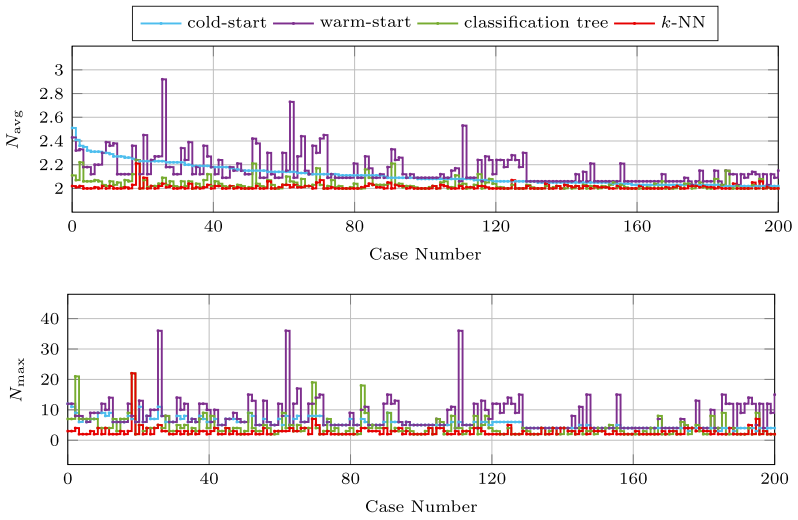


Fig. 4. Results for the Boeing 747 example: variable N_{avg} shows the average number of iterations, $N_{\max}$ is the maximum number of iterations per one closed-loop simulation. The closed-loop simulations involve the regulation of the Boeing 747 system (13) using MPC feedback policy.

as

$$A = \begin{bmatrix} 0.9993 & 0.0076 & -0.0005 & -0.0644 \\ -0.0096 & 0.9236 & 1.4290 & 0.0003 \\ 0.0039 & -0.0186 & 0.9034 & -0.0001 \\ 0.0004 & -0.0019 & 0.1907 & 1.0000 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.0019 & 0.1999 \\ -0.2051 & 0.0791 \\ -0.2208 & 0.1145 \\ -0.0225 & 0.0116 \end{bmatrix}, \tag{13a}$$

$$C = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & -1.0000 & 0.0000 & 7.7400 \end{bmatrix}. \tag{13b}$$

Next, an model predictive controller was designed with $N = 15$, matrices $Q_{\text{x}} = I_{4\times4}$, and $Q_{\text{f}} = I_{4\times4}$ were set to identity matrices, and to ensure the stability the input penalty was set to $Q_{\text{u}} = 10 \cdot I_{2\times2}$. Constraints on the manipulated variables were set to $[-3,\ -3]^{\intercal} \le u \le [3,\ 3]^{\intercal}$, while the measured variables were constrained to $[-42.5,\ -47.5]^{\intercal} \le y \le [27,\ 23]^{\intercal}$. The optimization problem consist of 30 decision variables and of 124 inequality constraints.

Similarly to previous case study, the training data were obtained by equidistant gridding of the feasibility domain. By this procedure, we obtained 122 initial conditions, from which we performed closed-loop simulations with 100 samples. The entire training set then consisted of 12 200 data tuples, with 190 unique combinations of active constraints. We constructed two classifiers with hyper-parameter optimization. The classification tree was constructed in 16 min and consisted of 295 nodes. The k-NN classifier was obtained in 4 min and is determined by 12 200 observations.

The performance of the CT-based and k-NN-based warm-starting of the ASM was tested on 200 simulations with initial conditions different from the training data set. Each simulation length was $N_{\text{sim}} = 100$ samples. Results of this comparison in shown in Fig. 4, where we compare the total number of iterations in each simulation case of the ML-enhanced ASM with standardized warm-starting and with the cold-start of the ASM. Again, we see that in 92.5% the usage of classification tree decreases the total number of iterations compared to the warm-starting of Ferreau et al. (2008), while the k-NN-based warm-start improves the iteration count in 98% of all cases.

6. Conclusions

In this paper, we have shown how to decrease the time to solve the MPC control problem posed as a QP. This was achieved by providing better estimates of the active constraints for the active-set method using machine learning tools. Specifically, the classification tree and the k-NN classifier, were used to supply an estimate of the set of active constraints based on the current value of the initial condition. The numerical case study showed, that the number of iterations required to converge to optimal solution can be reduced by several orders of magnitude.

Reduced number of iterations also decreases the time required to converge which allows for the embedded implementation of online MPC. Results gathered on hardware simulations showed reduced time of convergence when considering the machine-learning enhancements of the active set method. Moreover the hardware-in-the loop simulation shows that the time required to obtain the optimal control action is decreased by a factor of 20.

In our experiments, the k-NN classifier outperformed classification trees. Our explanation is that the classification tree uses axis-aligned separation hyperplanes, therefore the regions of validity of individual initial active sets are hyperboxes. However, from the theory of explicit MPC (Bemporad et al., 2002) we know that the exact regions of validity are generic polytopes, not necessarily hyperboxes. Therefore the k-NN classifier better approximates these general shapes, leading to an initialization of Algorithm 1 that is closer to the truly optimal one, hence allowing it to perform fewer iterations to converge.

Acknowledgments

The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grant no. 1/0403/15, the financial support of the Slovak Research and Development Agency under the project APVV-15-0007, and the Research & Development Operational Programme for the project University Scientific Park STU in Bratislava, ITMS 26240220084, supported by the Research 7 Development Operational Programme funded by the ERDF.

References

Axehill, D., Hansson, A., 2008. A dual gradient projection quadratic programming algorithm tailored for model predictive control. In: Decision and Control, 2008. CDC 2008. 47th IEEE Conference on. IEEE, pp. 3057–3064.

Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N., 2002. The explicit linear quadratic regulator for constrained systems. Automatica 38 (1), 3–20.

Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A., 1984. Classification and Regression Trees. In: The Wadsworth and Brooks-Cole statistics-probability series, Taylor & Francis.

Camacho, E.F., Bordons, C., 2007. Model Predictive Control, second ed. Springer.

Cimini, G., Bemporad, A., 2017. Exact complexity certification of active-set methods for quadratic programming. IEEE Trans. Automat. Control PP (99), 1–1.

Cristianini, N., Shawe-Taylor, J., 2000. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press.

Dantzig, G.B., Eaves, B.C., 1973. Fourier-motzkin elimination and its dual. J. Combin. Theory Ser. A 14 (3), 288–297.

Dasarathy, B.V., 1991. Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. In: IEEE Computer Society Press tutorial, IEEE Computer Society Press.

Dorfman, R., Samuelson, P., Solow, R., 1958. Linear Programming and Economic Analysis. Courier Corporation.

Elsayed, S.M., Sarker, R.A., Essam, D.L., 2014. A new genetic algorithm for solving optimization problems. Eng. Appl. Artif. Intell. 27, 57–69.

Ferreau, H., Bock, H., Diehl, M., 2008. An online active set strategy to overcome the limitations of explicit MPC. Internat. J. Robust Nonlinear Control 18 (8), 816–830.

Ferreau, H.J., Kirches, C., Potschka, A., Bock, H.G., Diehl, M., 2014. qpOASES: A parametric active-set algorithm for quadratic programming. Math. Program. Comput. 6 (4), 327–363.

Fletcher, R., 2013. Practical Methods of Optimization. Wiley.

Gurney, K., 2003. An Introduction to Neural Networks. Taylor & Francis.

Herceg, M., Jones, C.N., Morari, M., 2015. Dominant speed factors of active set methods for fast MPC. Optim. Control Appl. Methods 36 (5), 608–627.

Herceg, M., Kvasnica, M., Jones, C., Morari, M., 2013. Multi-parametric toolbox 3.0. In: 2013 European Control Conference. pp. 502–510.

Huyck, B., Ferreau, H.J., Diehl, M., De Brabanter, J., Van Impe, J.F.M., De Moor, B., Logist, F., 2012. Towards online model predictive control on a programmable logic controller: Practical considerations. Math. Probl. Eng. 2012.

Jones, C., Kerrigan, E., Maciejowski, J.M., 2004. Equality set projection: A new algorithm for the projection of polytopes in halfspace representation. Technical report, Cambridge University Engineering Department.

Kögel, M., Findeisen, R., 2011. A fast gradient method for embedded linear predictive control. IFAC Proc. Vol. 44 (1), 1362–1367.

Li, W., Swetits, J., 1997. A new algorithm for solving strictly convex quadratic programs. SIAM J. Optim. 7 (3), 595–619.

Ling, K.V., Wu, B.F., Maciejowski, J.M., 2008. Embedded model predictive control (MPC) using a FPGA. IFAC Proc. Vol. 41 (2), 15250–15255.

Liu, X., Kong, X., 2014. Nonlinear model predictive control for dfig-based wind power generation. IEEE Trans. Autom. Sci. Eng. 11 (4), 1046–1055.

Maciejowski, J.M., 2002. Predictive Control with Constraints. PEARSON Prentice-Hall.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M., 2000. Constrained model predictive control: Stability and optimality. Automatica 36 (6), 789–814.

Nazemi, A., 2014. A neural network model for solving convex quadratic programming problems with some applications. Eng. Appl. Artif. Intell. 32, 54–62.

Nocedal, J., Wright, S.J., 2006. Numerical Optimization, second ed. Springer, New York.

O'Donoghue, B., Stathopoulos, G., Boyd, S., 2013. A splitting method for optimal control. IEEE Trans. Control Syst. Technol. 21 (6), 2432–2442.

Otta, P., Santin, O., Havlena, V., 2015. Measured-state driven warm-start strategy for linear MPC. In: Control Conference (ECC), 2015 European. IEEE, pp. 3132–3136.

Pal, M., 2005. Random forest classifier for remote sensing classification. Int. J. Remote Sens. 26 (1), 217–222.

Patrinos, P., Bemporad, A., 2014. An accelerated dual gradient-projection algorithm for embedded linear model predictive control. IEEE Trans. Automat. Control 59 (1), 18–33.

Rao, C.V., Wright, S.J., Rawlings, J.B., 1998. Application of interior-point methods to model predictive control. J. Optim. Theory Appl. 99 (3), 723–757.

Richter, S., Jones, C.N., Morari, M., 2012. Computational complexity certification for real-time MPC with input constraints based on the fast gradient method. IEEE Trans. Automat. Control 57 (6), 1391–1403.

Rossiter, J.A., 2003. Model-based Predictive Control: A Practical Approach. CRC press.

Stathopoulos, G., Szucs, A., Pu, Y., Jones, C.N., 2014. Splitting methods in control. In: Control Conference (ECC), 2014 European. IEEE, pp. 2478–2483.

Wanxie, Z., Suming, S., 1988. A finite element method for elasto-plastic structures and contact problems by parametric quadratic programming. Internat. J. Numer. Methods Engrg. 26 (12), 2723–2738.

Wills, A., Ninness, B., 2010. Qpcquadratic programming in c. Software available at http://sigpromu.org/quadprog.

Wu, X., Liu, Y., 2012. Optimizing fuzzy portfolio selection problems by parametric quadratic programming. Fuzzy Optim. Decis. Mak. 11 (4), 411–449.

Zeilinger, M.N., Jones, C., Morari, M., 2011. Real-time suboptimal model predictive control using a combination of explicit MPC and online optimization. IEEE Trans. Automat. Control 56 (7), 1524–1534.

Zhang, H., 2004. The optimality of naive bayes. In: Barr, V., Markov, Z. (Eds.), Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004).

Zometa, P., Kögel, M., Findeisen, R., 2013. μao-MPC: a free code generation tool for embedded real-time linear model predictive control. In: American Control Conference (ACC), 2013. IEEE, pp. 5320–5325.

Martin Klauco

From: The Automatica Submission and Review Management System <automatica@ifac.papercept.net>
Sent: Thursday, February 24, 2022 17:08
To: juraj.oravec@stuba.sk
Cc: martin.klauco@stuba.sk
Subject: Automatica 21-0733, version 3: Publication decision
Attachments: ReportOn21-0733v3.txt

Message from The Automatica Submission and Review Management System

Decision: Accept as Technical Communique Editor's cover message:

Dear Dr. Oravec,

Your paper number 21-0733, submitted for publication in Automatica, has now been reviewed, and copies of the review reports are enclosed. It is my pleasure to inform you that, based on these reports and the advice of the Associate Editor concerned, your contribution is accepted for publication as a Technical Communique in Automatica.

Please prepare a final draft of your contribution to be submitted through the Automatica On-Line Review Management System available at www.autsubmit.com following the instructions available under the button "information for authors". The final draft of your paper should be resubmitted within 30 days of the date of this letter.

Your contribution to Automatica is very much appreciated. After we receive your final manuscript you will be informed on the disposition of your paper.

If you wish to contact editor André Tits or his editorial assistant Levi Burner concerning the present paper, please use the "Contact" menu option in Papercept's page for paper number 21-0733.

The publication decision and reviews may also be inspected on-line by logging in at

https://ifac.papercept.net/conferences/scripts/start.pl

If you are invited to submit a revised version of the paper or to submit the final version of the paper following acceptance then please log in and follow the appropriate link.

André L. Tits
Automatica Editor, Rapid Publications
Department of Electrical & Computer Engineering
and Institute for Systems Research
University of Maryland
College Park, MD 20742
tel: 301-405-3669, fax: 301-314-9920
Email: tits-automatica@umd.edu

To follow up on this message log in at
https://ifac.papercept.net/journals/automatica/ or through the portal site https://ifac.papercept.net/conferences with your PIN or Login alias and password, enter your workspace as Automatica author and check the action that is required from the corresponding author of this submission

Publication decision
==========
Accept

Resubmission deadline: March 26, 2022
Action expected from corresponding author:
Submit final version, preferably before the deadline

Submission Information
==========

Submission number
21-0733 (Submission ID 21-0733)

Submission title
Real-Time Tunable Approximated Explicit MPC

Authors
Juraj Oravec* (72850), Martin Klauco (72600)

Type of submission
Technical Communique

Version 3

Status
Accepted

Date of latest decision
February 24, 2022

Reply to
André Tits
Univ. of Maryland
tits-automatica@umd.edu, lburner@umd.edu

Real-Time Tunable Approximated Explicit MPC

Juraj Oravec, Martin Klaučo

Institute of Information Engineering, Automation, and Mathematics; Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava; Radlinskeho 9, SK812-37, Bratislava, Slovakia

Abstract

Tunability is a major obstacle in the creation and subsequent application of the explicit model predictive control (MPC). The main bottleneck lies in the need to reconstruct the parametric solution each time weighting factors changes. Such an operation makes the implementation of the explicit MPC impractical. This manuscript addresses the problem of producing a suboptimal parametric solution to the optimal control problem, where the change of the weighting factor does not warrant the reconstruction of the explicit MPC. The solution is achieved by interpolating between two boundary explicit solutions for a range of values in weighing factors. Furthermore, we show that the suboptimal solution enforces the closed-loop stability and recursive feasibility. The stability and recursive feasibility are maintained by carefully choosing the terminal penalty and terminal set in those boundary explicit solutions.

Key words: Model Predictive Control, Parametric Optimization, Tunable Controllers.

1 Introduction

In model predictive control (MPC) design, the tuning matrices serve a dual purpose. First is scaling individual components of the state and input vectors. For this purpose, the selection of penalty matrices is dictated by the physics of the plant. While in this scaling approach, we neglect the control-oriented objectives such as performance or comfort. The second purpose of tuning matrices is to manage the *aggressivity* of the controller. We can find many industrial applications where fixed values of tuning factors determined before control do not yield satisfactory performance over time, e.g., see [16,17,19], and references therein. Therefore, the tunability of the MPC is a highly demanded feature by control engineers. Implementation of optimal control problem with non-fixed tuning factors in explicit MPC framework [3] can be done in an optimal fashion in two ways. First, we can construct a parametric solution to the MPC problem subject to *all* parameters, which now includes tuning factors, resulting in non-linear parametric optimization. The second approach lies in reconstructing the explicit solution each time instant if weighting factors change, which counteracts the effect of having an explicit MPC. So far, a limited number of scientific works have been published in this direction. However, by addressing the issue of effective real-time implementation of tunable explicit MPC strategies, we significantly increase their implementation potential in various fields of applications [18]. Moreover, these controllers can, to some extent, replace gain-scheduling controllers with the added value of constraint satisfaction. The tunability properties of the explicit MPC solutions were previously addressed in [2], where a parametric solution with respect to the weighting factor has been derived. The drawback of the proposed solutions is its strict limitation to linear terms in the objective function, namely the $1/\infty$-norm. Furthermore, the authors in [2] provide a solution only for MPCs with scalar tuning factor multiplying the weighting factor related to the input penalty. Computing explicit MPC that can be re-tuned online will necessarily increase the complexity of the solution, as the number of parameters needs to be increased. Unfortunately, $1/\infty$-norm-based penalization setup is rarely used in practice, as the controller exhibits more aggressive behavior around the origin than the quadratic (2-norm) counterparts. Moreover, $1/\infty$-norm penalization induces a sort of a *dead-zone* behavior for the input penalty in the sense that despite the input penalty being tuned in some range, the optimal control input remains the same, i.e., the effect is discontinuous in contrast to the 2-norm case. A notable step forward in producing online tunable implementation of the explicit MPC with a 2-norm penalization was presented in [6]. Here the authors present an interpolation-based procedure, which allows for the online tuning of the input weighting factor leading to a suboptimal solution. The proposed approach relies on two

⋆ Corresponding author J. Oravec. Tel. +421 259 325 364
Email addresses: juraj.oravec@stuba.sk (Juraj Oravec), martin.klauco@stuba.sk (Martin Klaučo).

boundary explicit MPCs, which are constructed for two specific input penalties. The presented approach allows a real-time change in the input penalty matrix within a given range and allows modifying the penalty matrix during the operation without the necessity to resolve optimization problems. The idea is to *reconstruct* the close-to-optimal control action by devising the convex combination of the available solutions of two boundary explicit MPCs. The approach presented in this paper significantly improves the propositions in [6], as we generalize the original results and, under mild assumptions, we formulate a procedure where we guarantee the closed-loop system stability and recursive feasibility [13]. The proposed control strategy is significantly less complex compared to the conventional approach in [2], and the suboptimality level of the proposed scheme appears to be negligible.

2 Problem statement

Throughout the paper, we consider the following two MPC design problems implemented in the receding horizon control fashion [12]. The first boundary MPC formulations is stated as

$$\min_{u_0,u_1,\ldots,u_{N-1}} \quad x_N^\intercal P_\mathrm{L} x_N + \sum_{k=0}^{N-1} (x_k^\intercal Q_\mathrm{L} x_k + u_k^\intercal R_\mathrm{L} u_k) \tag{1a}$$

$$\text{s.t.:} \quad x_{k+1} = A\,x_k + B\,u_k, \tag{1b}$$

$$u_k \in \mathcal{U}, \tag{1c}$$

$$x_k \in \mathcal{X}, \tag{1d}$$

$$x_N \in \mathcal{T}_\mathrm{L}, \tag{1e}$$

$$x_0 = \theta(t), \tag{1f}$$

$$k = 0, 1, \ldots, N-1, \tag{1g}$$

while the second one has a modification in the objective function and terminal set, namely,

$$\min_{u_0,u_1,\ldots,u_{N-1}} \quad x_N^\intercal P_\mathrm{H} x_N + \sum_{k=0}^{N-1} (x_k^\intercal Q_\mathrm{H} x_k + u_k^\intercal R_\mathrm{H} u_k) \tag{2a}$$

$$\text{s.t.:} \quad (1\mathrm{b}), (1\mathrm{c}), (1\mathrm{d}), (1\mathrm{f}), (1\mathrm{g}), \tag{2b}$$

$$x_N \in \mathcal{T}_\mathrm{H}, \tag{2c}$$

where N is prediction horizon, $P_\mathrm{L}, P_\mathrm{H} \in \mathbb{R}^{n\times n}$, $Q_\mathrm{L}, Q_\mathrm{H} \in \mathbb{R}^{n\times n}$, $R_\mathrm{L}, R_\mathrm{H} \in \mathbb{R}^{m\times m}$, are terminal, state, and input pairs of the penalty matrices, respectively. Prediction model in (1b) has the form of linear time invariant (LTI) system for state matrix $A \in \mathbb{R}^{n\times n}$ and input matrix $B \in \mathbb{R}^{n\times m}$. Vectors $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ are vectors of system states and control inputs, respectively. $\mathcal{U} \subseteq \mathbb{R}^m$, $\mathcal{X} \subseteq \mathbb{R}^n$ are sets of input and state constraints, respectively. $\mathcal{T}_\mathrm{L}, \mathcal{T}_\mathrm{H} \subset \mathbb{R}^n$, are sets of terminal constraints. $\theta(t) \in \Omega$ is vector of initial conditions and $\Omega \subseteq \mathcal{X}$ is set of feasible initial conditions.

Assumption 2.1 *Let MPC problems* (1), (2) *be asymptotically stable and recursive feasible. Assume, in* (1), (2) *hold:*

(1) sets $\mathcal{U}$, $\mathcal{X}$, $\mathcal{T}_\mathrm{L}$, $\mathcal{T}_\mathrm{H}$, Ω are closed convex polyhedra containing origin in their strict interiors,

(2) matrices $P_\mathrm{L}, P_\mathrm{H}$, $Q_\mathrm{L}, Q_\mathrm{H}$, $R_\mathrm{L}, R_\mathrm{H}$ are positive definite.

As the initial condition in (1f) has a parametric form, the MPC problems in (1), (2) are problems of the multiparametric quadratic programming (mpQP), see [3].

Problem 2.1: Based on the parametric (*explicit*) solutions of the MPC problems in (1), (2), the task is to approximate the optimal solution $\widetilde{u}_0$ for *any* MPC problem having input or state penalty matrix *between* the matrix pair $(R_\mathrm{L}, R_\mathrm{H})$ or $(Q_\mathrm{L}, Q_\mathrm{H})$, respectively, without the necessity to solve the optimization problem.

3 Tunable explicit MPC

Problem 2.1 is addressed by approximated solution of the MPC problem having following form:

$$\min_{u_0,u_1,\ldots,u_{N-1}} \quad x_N^\intercal \widetilde{P} x_N + \sum_{k=0}^{N-1} \left(x_k^\intercal \widetilde{Q} x_k + u_k^\intercal \widetilde{R} u_k\right) \tag{3a}$$

$$\text{s.t.:} \quad (1\mathrm{b}), (1\mathrm{c}), (1\mathrm{d}), (1\mathrm{f}), (1\mathrm{g}), \tag{3b}$$

$$x_N \in \widetilde{\mathcal{T}}, \tag{3c}$$

where $\widetilde{P}$, $\widetilde{\mathcal{T}}$ are appropriate terminal penalty and terminal constraint set, respectively. For input and state penalty matrices in (3a), respectively, hold:

$$\widetilde{R} = (\rho - 1)R_\mathrm{L} + \rho\, R_\mathrm{H}, \quad 0 \le \rho \le 1, \tag{4a}$$

$$\widetilde{Q} = (\phi - 1)Q_\mathrm{L} + \phi\, Q_\mathrm{H}, \quad 0 \le \phi \le 1. \tag{4b}$$

Remark 3.1 (Linear tuning)
We consider a linear tuning, i.e., either input penalty $\widetilde{R}$ or state penalty $\widetilde{Q}$ in cost (3a) *is tuned. Therefore, if $0 < \rho < 1$ then $\widetilde{Q} = Q_\mathrm{L} = Q_\mathrm{H}$, and, vice-versa, if $0 < \phi < 1$ then $\widetilde{R} = R_\mathrm{L} = R_\mathrm{H}$ hold.*

Remark 3.2 (Input-penalty-based tuning)
Without loss of generality, hereafter, we consider a tuning of input penalty $\widetilde{R}$ according to (4a). *Analogous results hold for state penalty $\widetilde{Q}$ tuning according to* (4b).

Remark 3.3 (Application range) *The application range of the proposed tunable explicit MPC is not limited just to MPC formulation in* (3). *For the sake of simplicity, we consider the general regulation problem, but various formulations satisfying stability and recursive feasibility could be applied, e.g., robust [10], regionless [9], approximated [1], convex-lifting-based [15], etc.*

The general structure of the MPC problems in (1), (2) is restricted by following mild assumptions.

Assumption 3.1 *Consider following assumptions hold:*

(1) terminal penalties in (1a), (2a) *are same and computed according to $\widetilde{P} = P_\mathrm{L} = P_\mathrm{H}$ in* (3a),

(2) terminal sets in (1e), (2c) *are same and computed according to* $\widetilde{\mathcal{T}} = \mathcal{T}_{\mathrm{L}} = \mathcal{T}_{\mathrm{H}}$ *in* (3c),
(3) input penalties $R_{\mathrm{L}}, R_{\mathrm{H}}$ *are diagonal matrices such that* $\lambda_i(R_{\mathrm{L}}) \leq \lambda_i(R_{\mathrm{H}})$ *holds* $\forall i = 1, ..., m$, *where* λ *denotes vector of input penalty matrix eigenvalues.*

Remark 3.4 (Sufficient prediction horizon)
Assumption 3.1 is necessary to provide the guarantees on the closed-loop system stability and recursive feasibility. On the other hand, the sufficient length of the prediction horizon N lead to omitting Assumption 3.1(1), (2), see [14] to determine the minimum length prediction horizon providing the stability guarantees.

The main benefit of the real-time tunable explicit MPC is that for any $\rho \in [0, 1]$, the online evaluation of the approximated control action is optimization-free and boils down to a mere linear function evaluations.

Definition 3.1 (Approximated control input) *Given the parametric solutions of MPC problems in* (1), (2), *and current system measurement* $\theta(t) \in \Omega$. *The vector of approximated control inputs* $\widetilde{u} \in \mathbb{R}^m$ *for MPC problem in* (3) *is evaluated using the convex combination:*

$$\widetilde{u} = (\rho - 1)u_{\mathrm{L}} + \rho\, u_{\mathrm{H}}, \qquad 0 \leq \rho \leq 1, \tag{5}$$

where u_L, u_H, *respectively, are optimal control inputs of MPC problems* (1), (2) *for* $\theta(t)$.

Although the $\widetilde{u}_0$ is approximated solution, it provides primal feasibility of MPC problem in (3).

Lemma 3.5 (Primal feasibility) *Given parametric solution of MPC problems in* (1)–(2), *given MPC problem in* (3), *and state measurement* $\theta(t) \in \Omega$. *Approximated solution* $\widetilde{u}$ *evaluated by Definition 3.1 ensures primal feasibility of MPC problem in* (3) *for* $\theta(t)$.

Proof: Let $u_{\mathrm{L},k}$, $u_{\mathrm{H},k}$, $k = 0, 1, \ldots, N-1$, be the optimal solutions of MPC problems in (1)–(2), respectively. According to Definition 3.1, $\widetilde{u}$ is evaluated using (5). As the consequence, for any $\theta(t) \in \Omega$ holds:

$$\text{if } u_{\mathrm{L},k} \leq u_{\mathrm{H},k} \Rightarrow u_{\mathrm{L},k} \leq \widetilde{u}_k \leq u_{\mathrm{H},k}, \tag{6a}$$
$$\text{if } u_{\mathrm{L},k} \geq u_{\mathrm{H},k} \Rightarrow u_{\mathrm{H},k} \leq \widetilde{u}_k \leq u_{\mathrm{L},k}, \tag{6b}$$

for $\forall k = 0, 1, \ldots, N-1$.
Optimal solutions $u_{\mathrm{L},k}$, $u_{\mathrm{H},k}$ are primal feasible, i.e., $u_{\mathrm{L},k}, u_{\mathrm{H},k} \in \mathcal{U}$. From the definition of the convex set holds true that convex combination of $u_{\mathrm{L},k}, u_{\mathrm{H},k}$ evaluated by (5) satisfies that $\widetilde{u}_k \in \mathcal{U}$ for $\forall k = 0, 1, \ldots, N-1$. Analogous hold true for state constraints in (1d), (1e), as they are, according to a linear prediction model in (1b), a linear combination of control inputs $u_{\mathrm{L},k}$, $u_{\mathrm{H},k}$ for given state measurement $\theta(t) \in \Omega$. □

Assumption 3.2 (Terminal penalty)
Terminal penalty $\widetilde{P} = \widetilde{P}^{\intercal} \succ 0$ *in* (3a) *is a matrix of a common quadratic Lyapunov function* $V(x) = x^{\intercal}\widetilde{P}x$, $V : \mathbb{R}^n \to \mathbb{R}$, *constructed w.r.t. LTI prediction model in* (1b) *and tunable input penalty* $\widetilde{R}$ *in* (3a) *for* $\forall\rho$ *within the range of interval [0, 1] in* (4a).

In offline phase, there are various ways on how to construct and/or validate Lyapunov function candidates, and how to minimize their conservativeness. Inspired by the approach in [7], we evaluate the terminal penalty matrix $\widetilde{P}$ in (3a) by solving the following problem of semidefinite programming (SDP):

$$\min_{\gamma, X, Y} \quad \gamma + \mathrm{tr}(X) \tag{7a}$$

$$\text{s.t.:} \quad \begin{bmatrix} X & \star & \star & \star \\ AX + BY & X & \star & \star \\ Q^{\frac{1}{2}}X & 0 & \gamma I & \star \\ R_j^{\frac{1}{2}}Y & 0 & 0 & \gamma I \end{bmatrix} \succeq 0, \quad j = 1, 2, \tag{7b}$$

where decision variables are $X = X^{\intercal} \in \mathbb{R}^n$, $Y \in \mathbb{R}^{n \times m}$, and $\gamma \in \mathbb{R}$. $R_1 = R_{\mathrm{L}}$, $R_2 = R_{\mathrm{H}}$, and I, 0 are identity and zero matrices of appropriate dimensions, and symbol $\star$ denotes Hermitian structure of LMIs in (7b). Note, $X \succ 0$ is weighted inverted Lyapunov matrix such that $\widetilde{P} = \gamma\, X^{-1}$, and Y is well-known matrix of state feedback parametrization. Further technical details are introduced in [7]. The other strategies determining the terminal penalty are reported, for example, in [4,11].

Assumption 3.3 (Terminal constraint)
Terminal constraint set $\widetilde{\mathcal{T}}$ *is a maximal control invariant set* $\mathcal{C}_\infty$ *constructed in the intersection of the terminal penalties in* (1e), (2c) *w.r.t. input constraints in* (1c), *i.e., holds* $\widetilde{\mathcal{T}} = \mathcal{C}_\infty \subseteq (\mathcal{T}_{\mathrm{L}} \cap \mathcal{T}_{\mathrm{H}}) \subset \mathcal{X}$.

In offline phase, terminal sets $\mathcal{T}_{\mathrm{L}}$, $\mathcal{T}_{\mathrm{H}}$ in (1e), (2c) are determined to be control invariant for given MPC problems. We consider evaluation of these terminal sets based on the solution of matrix Riccati equations in the LQR-based control framework [5]. Next, following Assumption 3.3, the terminal set $\widetilde{\mathcal{T}}$ in (3c) is evaluated to be a maximal control invariant set $\mathcal{C}_\infty$ constructed in the intersection of $\mathcal{T}_{\mathrm{L}} \cap \mathcal{T}_{\mathrm{H}}$. $\mathcal{C}_\infty$ is constructed w.r.t. input constraints in (1c), i.e., once system states x_k enter $\mathcal{C}_\infty$, there always exists such control input $\widetilde{u}_k$ that ensures $x_{k+1} \in \mathcal{C}_\infty$ hold true. Note, $\mathcal{C}_\infty$ always exists, as there is a neighborhood of origin such that $0 \in \mathcal{T}_{\mathrm{L}}$, $0 \in \mathcal{T}_{\mathrm{H}} \Rightarrow \mathcal{T}_{\mathrm{L}} \cap \mathcal{T}_{\mathrm{H}} \neq \varnothing$. The iterative procedure constructing $\mathcal{C}_\infty$ is introduced, e.g., in [5].

Having a suitable terminal penalty matrix $\widetilde{P}$ and terminal constraint set $\widetilde{\mathcal{T}}$, we formulate the main results of this work.

Theorem 3.6 (Tunable explicit MPC)
Given parametric solutions of MPC problems in (1), (2), *given MPC problem in* (3), *current system state measurement* $\theta(t) \in \Omega$, *and corresponding control input* $\widetilde{u}_0$ *by Definition 3.1. If Assumptions 3.1,3.2, 3.3 hold, then the closed-loop system is asymptotically stable and problem in* (3) *is recursively feasible for any* ρ *within the range of interval [0,1] in* (4a).

Proof: According to Assumption 3.2, for $\forall\rho$ within the range of interval [0, 1] in (4a), there exists a common quadratic Lyapunov function represented by positive definite matrix $\widetilde{P}$ for MPC problem in (3). According to Assumption 3.1, the terminal constraint set $\widetilde{\mathcal{T}}$ and terminal penalty $\widetilde{P}$ of MPC problems in (1), (2) are the same. According to Assumption 3.3, terminal constraint set $\widetilde{\mathcal{T}}$ is maximal control invariant set $\mathcal{C}_\infty$ w.r.t. constraints on control inputs in (1c).
The pair of control inputs $u_{\mathrm{L},0}$, $u_{\mathrm{H},0}$ represent the optimal solutions of MPC problems in (1), (2), respectively. This pair of optimal control inputs $u_{\mathrm{L},0}$, $u_{\mathrm{H},0}$ was evaluated w.r.t. the same terminal penalty represented by $\widetilde{P}$, and system states converge to the same terminal penalty set $\widetilde{\mathcal{T}}$. As the consequence, they guarantees the closed-loop system stability and recursive feasibility for MPC problems in (1), (2), respectively.
According to Assumption 3.1, MPC problem in (3) share the same terminal penalty and terminal constraint set. According to Definition 3.1 and Lemma 3.5, for any $\widetilde{u}_0$ holds (6). As the consequence, $\widetilde{u}_0$ lead to closed-loop system stability and recursive feasibility of MPC problem in (3). □

Compared to the implementation of the conventional (non-tunable) explicit MPC, the proposed method increases effort in both: (i) offline phase by solving two parametric optimization problems and in (ii) online phase by evaluating two boundary explicit control laws. The evaluation of convex combination in (5) is negligible. Moreover, the memory footprint increases as the parametric solution of both MPC problems needs to be stored. On the other hand, the real-time tunable explicit MPC offers an *infinite* number of stabilizing sub-optimal control actions (i.e., MPC controllers) for input penalty tuning given by $\rho \in [0, 1]$ (or $\phi \in [0, 1]$ for state penalty tuning) in (4) without the necessity to solve any optimization problem. This valuable benefit enables convenient tuning, validation, and verification of the closed-loop control performance reducing the evaluation effort to a few linear algebra operations. The increased online effort could be reduced by introducing advanced techniques recalling some information from the computation of u_{L} to speed up evaluation of u_{H} in (5). This approach goes beyond the scope of this paper and is a subject of further research.

Remark 3.7 (1-norm and ∞-norm cost)
Cost functions in MPC problems (1), (2) *have the form of (squared)* 2*-norm. Analogous results of Lemma 3.5 and Theorem 3.6 hold for 1-norm and* ∞*-norm cost functions. Moreover, as 1-norm and* ∞*-norm are of linear nature (piecewise affine), the suboptimality level of approximated control input in* (5) *converge much faster to the optimum value for given prediction horizon* N*. The closed-loop system stability and recursive feasibility issues of Assumptions 3.2, 3.3 can be addressed according to Remark 3.4.*

4 Example

The simplified numerical example demonstrates the real-time tunable level of *aggressiveness/energy loses* of the closed-loop performance under different setups of (Q, R) proportion. This ability is illustrated considering a well-known example of a double integrator system with LTI prediction model in (1b) having $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix}$, and following setup of MPC problems in (1), (2): $Q_{\mathrm{L}} = Q_{\mathrm{H}} = I$, $R_{\mathrm{L}} = 0.5$, $R_{\mathrm{H}} = 10.0$, $\mathcal{U} = \{u : -1 \leq u \leq 1\}$, $\mathcal{X} = \{x : -5 \leq x \leq 5\}$, and $N = 2, 3, 5, 10$. Solving SDP in (7) and evaluation of $\mathcal{C}_\infty$, return terminal penalty and terminal set in the form

$$\widetilde{P} = \begin{bmatrix} 6.3743 & 0.5172 \\ 0.5172 & 15.5601 \end{bmatrix}, \quad \widetilde{\mathcal{T}} = \left\{ x : \begin{bmatrix} 0.0651 & -0.4463 \\ 0.1742 & -0.2388 \\ 0.4004 & 0.6479 \\ -0.0651 & 0.4463 \\ -0.1742 & 0.2388 \\ -0.4004 & -0.6479 \end{bmatrix} x \leq \begin{bmatrix} 0.8925 \\ 0.9553 \\ 0.6479 \\ 0.8925 \\ 0.9553 \\ 0.6479 \end{bmatrix} \right\}.$$

Based on value of N, solving [1] MPC problems (1), (2) lead to polytopic partitions having from 9 up to 21 critical regions for (1) and from 11 up to 15 critical regions for (2), after processing an optimal regions merging [8]. The series of explicit MPC controllers for input penalty $\widetilde{R}$ corresponding to particular setup of tuning parameter $\rho = 0.25, 0.50, 0.75$ in (5) was constructed to investigate the closed-loop performance. Figure 1 depicts the closed-loop control trajectories excited by initial conditions $x_0 = x_{15} = x_{30} = [4.5, -2.7]^\intercal$ for particular setup of $N = 10$. The control profiles generated by approximated control inputs $\widetilde{u}$ are compared to optimal performance u_{opt}. Table 1 summarizes construction time t_{con} of explicit controllers and the average real-time evaluation of optimal t_{opt} and approximated t_{app} control actions running non-optimized code on a non-industrial hardware. It can be observed that approximated control action increased a runtime, approximately, by factor 2. On the other hand, there is no need to solve any multi-parametric optimization problem online. Moreover, the performance loss computed as a ratio of optimal and approximated closed-loop costs is negligible, as reported in Table 1.

Table 1
Performance criteria.

	performance loss [%]			offline [s]	online [ms]	
N	$\rho = 0.25$	$\rho = 0.50$	$\rho = 0.75$	t_{con}	t_{opt}	t_{app}
2	0.23	0.13	0.11	0.34	0.52	1.07
3	0.36	0.27	0.10	0.45	0.55	1.12
5	0.26	0.28	0.16	0.80	0.57	1.13
10	0.21	0.22	0.13	1.26	0.63	1.27

[1] The results were generated using Intel(R) Core(TM) i7-1065G7 CPU 1.50 GHz, 16 GB RAM, MATLAB R2021b, YALMIP R20210331, MPT v3.2.1, MOSEK v9.3.6.

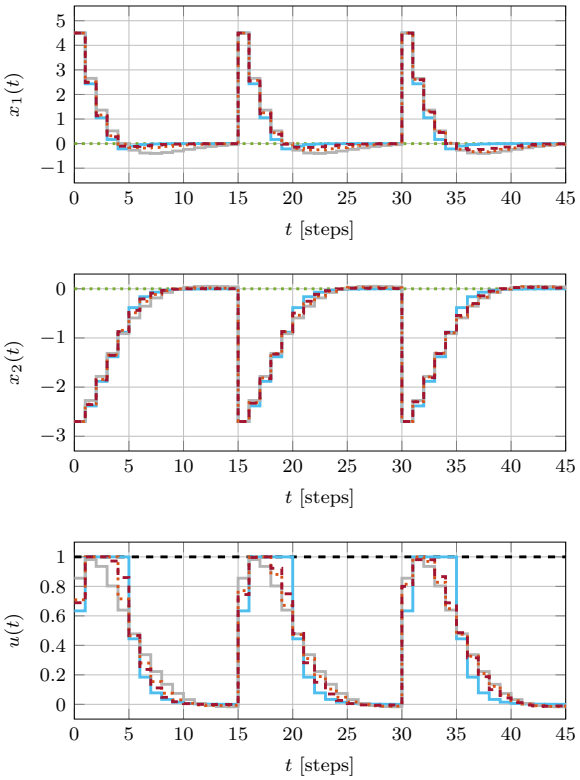


Fig. 1. Control performance of real-time tunable explicit MPC ensured by $\widetilde{u}$ (dark red, dashed), u_{L} (blue, solid), u_{H} (grey, solid), u_{opt} (orange, dotted), reference (green, dotted), constraints (black, dashed) for sequence of tuning parameter $\rho = 0.25, 0.50, 0.75$.

Acknowledgements

The authors sincerely thank Michal Kvasnica, whose comments helped improve the manuscript. The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grants 1/0585/19, 1/0297/22, and the Slovak Research and Development Agency under the project APVV-20-0261.

References

[1] P. Bakaráč, J. Holaza, M. Klaučo, M. Kalúz, J. Löfberg, and M. Kvasnica. Explicit MPC based on approximate dynamic programming. In *ECC 2018*, pages 1172–1177, Limassol, Cyprus, 2018.

[2] M. Baric, M. Baotic, and M. Morari. On-line tuning of controllers for systems with constraints. In *Proc. of the 44th IEEE CDC*, pages 8288–8293, 2005.

[3] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38:3–20, 2002.

[4] H. H. J. Bloemen, T. J. J. van den Boom, and H. B. Verbruggen. Optimizing the end-point state-weighting matrix in model-based predictive control. *Automatica*, 38(6):1061–1068, 2002.

[5] F. Borrelli. *Constrained Optimal Control of Linear and Hybrid Systems*. Springer Berlin Heidelberg, 2017.

[6] M. Klaučo and M. Kvasnica. Towards on-line tunable explicit MPC using interpolation. In *Preprints of the 6th IFAC Conference on Nonlinear Model Predictive Control*, Madison, Wisconsin, USA, 19-22.8 2018.

[7] M. V. Kothare, V. Balakrishnan, and M. Morari. Robust Constrained Model Predictive Control Using Linear Matrix Inequalities. *Automatica*, 32:1361–1379, 1996.

[8] M. Kvasnica, J. Holaza, B. Takács, and D. Ingole. Design and verification of low-complexity explicit MPC controllers in MPT3. In *ECC 2015*, pages 2600–2605, Linz, Austria, 2015.

[9] M. Kvasnica, B. Takács, J. Holaza, and S. Di Cairano. On region-free explicit model predictive control. In *54th IEEE CDC*, pages 3669–3674, Osaka, Japan, 2015.

[10] M. Kvasnica, B. Takács, J. Holaza, and D. Ingole. Reachability analysis and control synthesis for uncertain linear systems in MPT. In *Proc. of the 8th IFAC ROCOND*, pages 302–307, Bratislava, Slovakia, 2015.

[11] J.W. Lee, W. Hyun Kwon, and J. Choi. On stability of constrained receding horizon control with finite terminal weighting matrix. *Automatica*, 34(12):1607–1612, 1998.

[12] J. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, London, 2000.

[13] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[14] M. Mönnigmann. On the structure of the set of active sets in constrained linear quadratic regulation. *Automatica*, 106:61–69, 2019.

[15] N. A. Nguyen, M. Gulan, S. Olaru, and P. Rodriguez-Ayerbe. Convex lifting: Theory and control applications. *IEEE Trans. on Automatic Control*, 63(5):1243–1258, 2018.

[16] J. De Schutter, M. Zanon, and M. Diehl. TuneMPC–a tool for economic tuning of tracking (N)MPC problems. *IEEE Control Systems Letters*, 4(4):910–915, 2020.

[17] F. Sorourifar, G. Makrygirgos, A. Mesbah, and J. A. Paulson. A data-driven automatic tuning method for MPC under uncertainty using constrained Bayesian optimization. *IFAC-PapersOnLine*, 54(3):243–250, 2021. 16th IFAC Symposium on Advanced Control of Chemical Processes.

[18] J. Theunissen, A. Sorniotti, P. Gruber, S. Fallah, M. Ricco, M. Kvasnica, and M. Dhaens. Regionless explicit model predictive control of active suspension systems with preview. *IEEE Trans. on Industrial Electronics*, 67(6):4877–4888, 2020.

[19] W. Wojsznis, J. Gudaz, T. Blevins, and A. Mehta. Practical approach to tuning MPC. *ISA Transactions*, 42(1):149–162, 2003.

Advances in Industrial Control

Martin Klaučo
Michal Kvasnica

MPC-Based Reference Governors

Theory and Case Studies

AIC
Springer

Advances in Industrial Control

Advances in Industrial Control is a series of monographs and contributed titles focusing on the applications of advanced and novel control methods within applied settings. This series has worldwide distribution to engineers, researchers and libraries.

The series promotes the exchange of information between academia and industry, to which end the books all demonstrate some theoretical aspect of an advanced or new control method and show how it can be applied either in a pilot plant or in some real industrial situation. The books are distinguished by the combination of the type of theory used and the type of application exemplified. Note that "industrial" here has a very broad interpretation; it applies not merely to the processes employed in industrial plants but to systems such as avionics and automotive brakes and drivetrain. This series complements the theoretical and more mathematical approach of Communications and Control Engineering.

Indexed by SCOPUS and Engineering Index.

Proposals for this series, composed of a proposal form downloaded from this page, a draft Contents, at least two sample chapters and an author cv (with a synopsis of the whole project, if possible) can be submitted to either of the:

Series Editors

Professor **Michael J. Grimble**
Department of Electronic and Electrical Engineering, Royal College Building, 204 George Street, Glasgow G1 1XW, United Kingdom
e-mail: m.j.grimble@strath.ac.uk

Professor **Antonella Ferrara**
Department of Electrical, Computer and Biomedical Engineering, University of Pavia, Via Ferrata 1, 27100 Pavia, Italy
e-mail: antonella.ferrara@unipv.it

or the

In-house Editor

Mr. **Oliver Jackson**
Springer London, 4 Crinan Street, London, N1 9XW, United Kingdom
e-mail: oliver.jackson@springer.com

Proposals are peer-reviewed.

Publishing Ethics

Researchers should conduct their research from research proposal to publication in line with best practices and codes of conduct of relevant professional bodies and/or national and international regulatory bodies. For more details on individual ethics matters please see: https://www.springer.com/gp/authors-editors/journal-author/journal-author-helpdesk/publishing-ethics/14214.

More information about this series at http://www.springer.com/series/1412

Martin Klaučo · Michal Kvasnica

MPC-Based Reference Governors

Theory and Case Studies

Springer

Martin Klaučo
Institute of Information Engineering,
Automation, and Mathematics
Slovak University of Technology
in Bratislava
Bratislava, Slovakia

Michal Kvasnica
Institute of Information Engineering,
Automation, and Mathematics
Slovak University of Technology
in Bratislava
Bratislava, Slovakia

ISSN 1430-9491 ISSN 2193-1577 (electronic)
Advances in Industrial Control
ISBN 978-3-030-17404-0 ISBN 978-3-030-17405-7 (eBook)
https://doi.org/10.1007/978-3-030-17405-7

Mathematics Subject Classification (2010): 49J15, 93C83, 93C05, 93B52

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To my parents, Katarína and Branislav
To my wife Katrin and my son Michal

Series Editor's Foreword

The aim of the series *Advances in Industrial Control* is to fill the gap between theoretical research and practical applications in the area of control engineering, contributing to the transition into practice of the most advanced control results, and bearing in mind significant recent developments of control technology. It also promotes the dissemination of knowledge related to the modern control solutions in all sectors of industrial control, making it usable even to readers who are not experts in the specific application field.

One of the distinctive features of the series *Advances in Industrial Control* is the large variety of control methodologies which are reviewed and discussed with reference to different kinds of industrial applications. Some of the control methodologies discussed are extremely new and are explored in terms of their advantages and limitations in industrial implementations for the first time. Others are more classical but revisited in a more up-to-date fashion, taking into account the recent technological developments that make their use in practical applications nonetheless new.

The present monograph is indeed dedicated to a rather classical control topic, the so-called reference governor approach. It first appeared in the 90s in the continuous-time framework and in connection with linear dynamic systems. It was further developed and extended to more general classes of systems in the subsequent decades. A reference governor is typically used to modify the reference signal to a low-level control scheme in order to satisfy state and control constraints. Nevertheless, it can be used to generate a reference signal which satisfies some optimality requirements, or, in other cases, to improve an already-existing control system avoiding an expensive upgrade of the control infrastructure.

Although the topic can be considered classical, it is treated in the book in an original and in-depth way. What differentiates the present monograph from the previous publications on the same subject is the industrial cutting. When using the reference governor approach in practice, especially in the field of complex industrial plants, it is not at all obvious how to design the governor itself. The modeling of the dynamics of the closed-loop system on which the reference governor is inserted becomes a crucial node. This is particularly true when the generation of the

reference signal is made using optimization techniques. In fact, the approach must be implementable using the technology that is normally present in the industrial sector. It must therefore be reviewed considering computation and implementation issues. This and other aspects related to the use of the reference governor approach in industry are fully dealt with in this book.

The particular attention that this book devotes to implementation and technological aspects of industrial control is confirmed by the care that has been put into developing four interesting case studies related to realistic systems: a multivariable boiler–turbine power plant, a magnetic levitation system, a residential building, and a cascade of chemical reactors. In all the cases, the authors clarify how the process performance can be significantly increased using the reference governor approach, while providing safety guarantees by design.

This monograph enriches the series *Advances in Industrial Control* with a volume which is likely to be very useful, first of all, to students and researchers, since they will find in it a thorough tutorial overview of a classical control technique that is very effective in practice. It will also be interesting for practitioners, who will have precise indications and examples on how to transfer this approach into real industrial control situations. Like most of the books in the series *Advances in Industrial Control*, apart from its usefulness, it is also pleasant to read, with a fair balance between the theoretical part and the application-oriented part.

Pavia, Italy

Antonella Ferrara
University of Pavia

Preface

Control design for complex cyberphysical systems is a challenging task, especially when the safety and economic performance of the control system are considered. Two approaches are typically followed. The first option is to split the controlled process into subsystems, each operated by its own feedback loop, giving rise to a hierarchical composition of multiple controllers. The main issue, however, is how to configure and coordinate individual controllers such that they achieve a shared goal, such as the satisfaction of process constraints (e.g., pressures, temperatures, concentrations, voltages, currents, etc.) and optimization of the economic performance (e.g., minimization of the consumption of raw materials and heat/mass flows or maximization of the purity and the yields of products). This hierarchical direction therefore requires significant resources to be invested towards finding a suitable tuning of individual controllers and to perform extensive tests to confirm the satisfaction of the design goals. On the plus side, the control system being split into components allows individual subsystems to operate autonomously, thus providing easy maintenance and upgrade.

The second option is to employ a model-based control design procedure where the process is operated by a single, centralized, control system that takes all interactions into account. On the one hand, such an approach yields a safe and economically optimal operation by design as it is based on determining the control actions by solving an optimization problem that explicitly accounts for process constraints and economic performance. On the other hand, however, the centralized nature poses significant challenges in terms of practical implementation. Specifically, the centralized control system represents a single point of failure and any disruption of the main controller can render the whole production processes inoperable. Moreover, the centralized controller requires significant implementation resources in terms of computational power and storage as to be able to calculate control moves for the whole system at once.

This monograph covers a third direction, referred to as the *reference governor* approach. It can be viewed as an extension of the hierarchical design procedure where the control system is composed of numerous low-level feedback loops coordinated by a central model-based node. Its task is to determine, in a safe and

economically effective fashion, the setpoints for the low-level controllers. Such an arrangement provides several crucial advantages. First, safety and maximization of the economic performance are achieved by design as the selection of the setpoints is performed by solving an optimization problem that directly takes these objectives into account. Second, keeping the low-level controllers in place abolishes the single point of failure, allowing the production to continue even if there is disruption (e.g., during upgrades and maintenance) in the central node. Third, the reference governor approach is an ideal solution for retrofit applications where the aim is to improve the quality and the quantity of the production without having to invest significant resources into a costly upgrade of the control infrastructure. Last but not least, the reference governor framework allows for a separation of timescales where the central decision-maker needs can run on a longer sampling time (thus reducing the required computational resources), delegating the handling of fast dynamics to the simple low-level controllers.

Several questions arise when employing the reference governor framework in practice. The first one is how to capture the inherent dynamics of the low-level controllers in a way that allows for subsequent optimization of setpoints. Conventionally, reference governor setups have only been employed when the underlying low-level controllers are linear, as is the case of PID or LQR controllers. The reason being that the subsequent calculation of optimal setpoints boils down to solving a relatively simple optimization problem (denoted as a *convex* problem), allowing for its real-time implementation on the existing control hardware. However, it is far from clear how to apply reference governors if the low-level regulators are not linear. This monograph provides one of the possible answers. Specifically, it shows how to formulate the search for optimal setpoints for two important classes of low-level controllers: rule-based regulators and optimization-based controllers. In the rule-based scenario, the low-level controllers are represented by finite state machines that entail heuristic switching laws, such as those conveniently used in thermostats. The second category is represented by low-level controllers that use model predictive control to determine the optimal control inputs. For either case, the monograph shows that the search for optimal setpoints boils down to solving mixed-integer optimization problems.

The second question is how to formulate the central decision-making optimization problem in a way that allows it to be solved swiftly on hardware that is typically used in process automation. Two avenues are persuaded. The first one is based on pre-calculating the optimal solution of the reference governor problem off-line using *parametric programming*. Once available, the solution can then be encoded as a lookup table, allowing for a very fast and simple implementation in the real-time environment even on hardware with modest computational resources. To mitigate the storage requirements of parametric solutions, the lookup table can be further postprocessed to reduce its size at the expense of a slight deterioration of performance, but without sacrificing safety. The second avenue concentrates on selecting the setpoints for low-level controllers by solving an optimization problem online at each sampling instance. Here, the monograph shows various ways of

achieving a formulation that is easy and fast to solve even if the underlying optimization problem features integer variables.

The final question is how to demonstrate versatility and applicability of the reference governor approach. To this respect, the monograph discusses four realistic case studies. The first one, reported in Chap. 7, considers the optimization of setpoints for multiple PID controllers that control a multivariable boiler–turbine powerplant. In Chap. 8, the control of a magnetic levitation system with a single PID low-level controller is introduced. The controlled system features a fast dynamics with response times in the order of milliseconds and therefore necessitates a fast implementation of the reference governor. This is achieved by resorting to parametric solutions encoded as lookup tables. Implementation of a reference governor for a system with rule-based low-level controllers is reported in Chap. 9. The case study considers the management of thermal comfort in residential buildings where the reference governor optimizes temperature setpoints for a relay-based thermostat. Finally, the design of a reference governor for multiple optimization-based low-level controllers for a cascade of chemical reactors is discussed in Chap. 10. All case studies illustrate that with sufficient care, the process performance can be significantly increased while providing safety guarantees by design.

The objectives of the monograph can be summarized as follows:

- introduction to mathematical optimization problems required to determine optimal setpoints (Chap. 2),
- formulation of model-based optimal decision-making problems via the model predictive control framework (Chap. 3),
- description of modeling techniques for various types of low-level controllers (Chaps. 4–6),
- demonstration of the versatility and applicability of the reference governor framework by means of realistic case studies (Chaps. 7–10).

The reported results summarize and extend the research outcomes of the authors since 2013 that have been published in numerous peer-reviewed journals and presented at various IFAC and IEEE conferences. The monograph is structured as a step-by-step guide that allows theoreticians and practitioners alike to understand the underlying mathematical concept, to formulate reference governor solutions, and to implement them in practice.

Bratislava, Slovakia — Martin Klaučo
Michal Kvasnica

Acknowledgements

We want to acknowledge the financial aid of several agencies, which have supported the research covered in this book. Specifically, the authors gratefully acknowledge the contribution of the Slovak Research and Development Agency under the projects APVV-15-0007 (Optimal Control for Process Industries) and APVV-19-0204 (Safe and Secure Process Control), and the contribution of the Scientific Grant Agency of the Slovak Republic under the grant 1/0585/19 (On-Line Tunable Explicit Model Predictive Control for Systems with a Fast Dynamics). Moreover, the research was partly supported by the project ITMS 26240220084 (University Scientific Park STU in Bratislava) that was funded via the Research 7 Development Operational Programme funded by the ERDF, by the Alexander von Humboldt Foundation grant AvH-1065182-SVK (Embedded Optimal Control), and M. Klaučo would like to thank for the financial contribution from the STU in Bratislava Grant Scheme for Excellent Research Teams (Economically Effective Control of Energy Intensive Chemical Processes).

We want to acknowledge also the support of our colleagues, especially Martin Kalúz, who helped with the hardware implementation and experiments covered in Chap. 8, Ján Drgoňa for insights about the thermal comfort control needed in Chap. 9, and Juraj Holaza for helping with the results discussed in Chap. 10. Our thanks also go to Miroslav Fikar, the head of the Institute of Information Engineering, Automation, and Mathematics at the Slovak University of Technology, who created a challenging and yet comfortable working environment.

Bratislava, Slovakia — Martin Klaučo
February 2019 — Michal Kvasnica

Contents