Development of a Python tool based on model predictive control for an optimal management of the Calais canal

Pour, Fatemeh Karimi; Duviella, Eric; Segovia, Pablo

DOI
10.1016/j.ifacol.2022.11.001

Publication date
2022

Document Version
Final published version

Published in
IFAC-PapersOnline

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 55-33 (2022) 1–6

Development of a Python tool based on model predictive control for an optimal management of the Calais canal ⋆

Fatemeh Karimi Pour * **Eric Duviella** * **Pablo Segovia** **

* *IMT Nord Europe, Institut Mines-Télécom, Centre for Digital Systems, F-59000 Lille, France (e-mail: fatemeh.k.p@gmail.com, eric.duviella@imt-nord-europe.fr)*

** *Department of Maritime and Transport Technology, Delft University of Technology, Delft, the Netherlands (e-mail: p.segoviacastillo@tudelft.nl)*

Abstract: Model predictive control (MPC) has been widely employed to control a large variety of water systems, such as dams, irrigation canals, inland waterways, drinking water networks and wastewater treatment plants. Its predictive capabilities and the possibility to incorporate constraints make MPC well suited to address several, and sometimes opposite, management objectives linked to water systems. The design of MPC for water systems is usually performed via dedicated software (e.g., Matlab) and tested in simulation using dedicated hydraulic software. However, the implementation of MPC strategies in real systems requires additional development to allow for its embedding within the information systems that are used by system managers. A possible solution is to create a tool based on Python that can be easily integrated with the information systems of managers, and within which existing Matlab solutions can be incorporated. In this paper, the development a ready-to-use Python tool using a hierarchical MPC approach designed for the management of the Calais Canal is presented.

Keywords: Water systems, large-scale systems, model predictive control, hierarchical control, Python, Calais canal.

1. INTRODUCTION

Model predictive control (MPC) is a predictive control strategy that uses a process model to determine the control actions that yield optimal system performance with respect to a set of operational objectives while respecting constraints (Camacho and Bordons, 2013). MPC has been used in different applications including water systems, which are large-scale networked systems consisting of multiple interacting components, and which used to answer human's needs in terms of irrigation, transport, water supply, energy generation, industrial production, sanitation and leisure. Several paper deal with the predictive control of irrigation networks (Nguyen et al., 2017; Zheng et al., 2019; Guo and You, 2019; Chen et al., 2021), drinking water networks (Grosso et al., 2017; Baunsgaard et al., 2017; Karimi Pour et al., 2018; Karimi Pour and Puig, 2021; Segovia et al., 2021), inland waterways (Horváth et al., 2015; Segovia et al., 2017, 2019; Guekam et al., 2021) and sewer networks (Svensen et al., 2021; Sun et al., 2021; Sheik et al., 2021).

Segovia et al. (2020) design a hierarchical MPC to regulate the water levels in inland waterways taking into account the discrete states of the system, which are defined by the switching between low and high tides. Tidal periods are taken into account at the higher layer to modify the constraints and the hydraulic devices that can be used: while outlet sea gates can be opened to discharge water by gravity during low tide, the gates are no longer available during high tide, and only pumps can be used. However, the use of pumps leads to high electricity costs. Hence, an optimal control strategy should minimize the use of pumps by playing with the tidal range of the canal, i.e., accumulate water in the canal during high tide and release it during the next low tide period using the gates. At the same time, the control strategy must incorporate the main management objective, i.e., avoid flooding by keeping the level of the canal below a safety limit. The MPC strategy has been designed and validated using a simulation architecture between Matlab [1] and SIC^2 [2] software. Two hydraulic softwares have been investigated: on the one hand, HEC-RAS [3], which allows to use a dynamical model of canals using an accurate topographical and technical survey (Deshays et al., 2021); on the other hand, SIC^2, for which digital twins of real canals have been created (Ranjbar et al., 2020). The Matlab-SIC^2 architecture has been found to be more appropriate in terms of computational burden, and has been used for the canal of Calais located in the north of France.

The control strategy is designed to be used in real time to assist the Calais canal managers. Given the fact that the MPC developed in Matlab cannot be directly used in real time without the right license, a solution consisting in coding the MPC strategy in Python [4] is proposed. To this end, it is necessary to (i) install all the required libraries, (ii) transcode the Matlab files into Python, (iii) synchronize the execution of the code with the information systems used by managers, and (iv) provide optimal control values to the managers. There are some papers in the literature dealing with similar problems. Takács et al.

⋆ This work has been supported by ARMINES through the contract n° 1908V/1700661 and the IIW (Institution Intercommunale des Wateringues).

[1] https://www.mathworks.com
[2] http://sic.g-eau.net/?lang=en
[3] https://www.hec.usace.army.mil/software/hec-ras/
[4] https://www.python.org/

10.1016/j.ifacol.2022.11.001

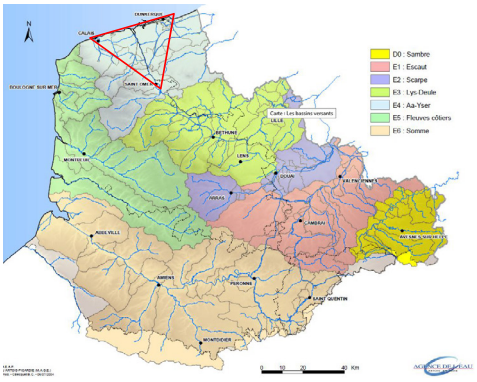

Fig. 1. Map of the watersheds in northern France and the *Wateringues* territory (red triangle)

(2015) implemented an MPC in Python, and demonstrate its suitability to control a quadrocopter and an artificial player in a video-game. Gehbauer (2019) build a package as a back-end for multi-layer and multi-time domain controller, with an application to weather forecast. Therefore, Python can be used to build simulation environments.

The main contribution of this paper is the transcoding of a hierarchical MPC approach (originally developed in Matlab) into Python for its embedding within the information system of the Calais canal managers. The main objective is to allow for a real-time deployment of the the hierarchical MPC, which allows to determine control actions for gates and pumps taking the time-varying tidal periods into account, and reject unknown disturbances.

The rest of the paper is organized as follows: the management of the Calais canal is described in Section 2. The design of the hierarchical MPC is detailed in Section 3. Transcoding of the hierarchical MPC into Python and its embedding in the information system are presented in Section 4. Conclusions and future research are discussed in Section 5.

2. MANAGEMENT OF THE CALAIS CANAL

The Calais canal is located in the north of France and belongs to the *Wateringues* territory, which is characterized by lowlands in maritime plains below high sea level [5] and forms a triangle with an area of 100,000 hectares (*see* Figure 1). The Calais canal is mainly used for navigation purposes, but also to release the excess of water in the waterways into the sea. It is principally supplied by pumping stations (PS) that are controlled by the *Wateringues* sections, and also by the canal upstream and three secondary canals along its course, namely the *Audruicq*, *Ardres* and *Guines* canals.

The canal is 28 km long and stretches from the Hennuin lock to the sea at Calais (*see* Figure 2). It is equipped with 18 PS, although only four of them are monitored, i.e., *Nouvelle Eglise*, *Les Attaques*, *Balinghem* and *3 Cornets*. Moreover, two outlet sea gates can be opened to release water into the sea by gravity. In addition to the gates, four pumps (two in Calais and two in Batellerie) can be used to dispose of water excess. Furthermore, the canal is equipped with three level sensors: one upstream and close to the Hennuin lock, another 8 km away from Calais and

[5] http://www.floodcom.eu/

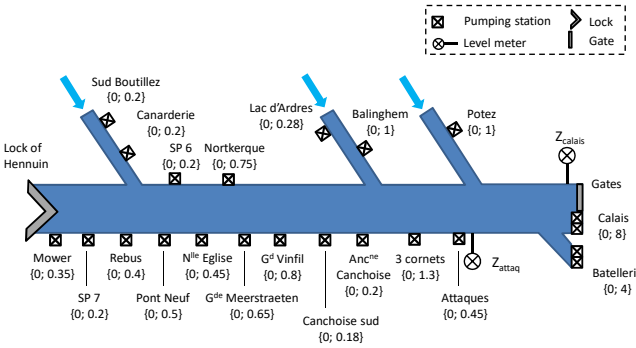

Fig. 2. Schematic representation of the Calais canal

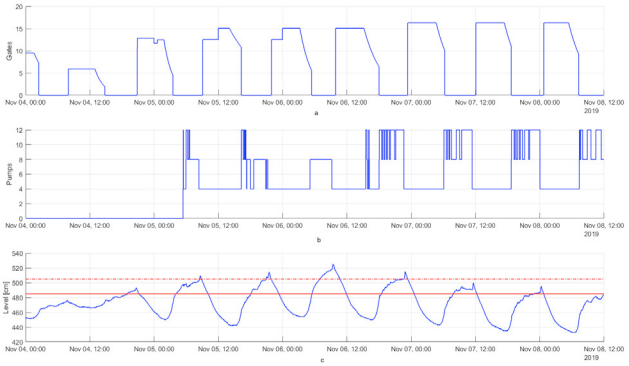

Fig. 3. Data corresponding to the Calais canal (Nov 4 to Nov 8). (a) Control of sea outlet gates. (b) Control of pumps. (c) Water level.

close to *Les Attaques*, and another downstream in Calais. An additional sensor provides sea level measurements.

Figure 3 depicts real data sequences corresponding to the Calais canal management, recorded from November 4, 2019 to November 8, 2019. It can be observed that gate opening increased until delivering the maximum discharge values (*see* Figure 3a). The gates cannot be opened during low tide, and this change in terms of tidal period can be easily noticed. Pumps, whose usage is not constrained by the tide, were first switched on in November 5 to limit the canal level increase (*see* Figure 3b). The level oscillated according to the control actions, but flooding episodes were not avoided (*see* Figure 3.c). The high navigation limit and the flooding limit are depicted in red solid and red dashed lines in Figure 3c, respectively. This episode constitutes an illustrative example of the operational objectives and constraints linked to the management of the Calais canal.

The management of the Calais canal is based on Logic Control Regulation (Duviella and Hadid, 2019), i.e., *if-then-else* rules. This type of control leads to a satisfactory management, except for some extreme rainy events. Therefore, an MPC-based strategy has been investigated to improve operational performance, and is presented in the next section. However, it must be noted that the optimal control actions are provided to managers as guidelines, who then decide whether to apply them.

3. DESIGN OF THE PREDICTIVE CONTROL STRATEGY

3.1 State-space model

The discrete-time control-oriented inland waterway model, proposed by Segovia et al. (2019), is simplified by considering a

unique delay τ. This assumption can be made because a single delay is sufficient to characterize the system dynamics. The model is described for all time instant $k \in \mathbb{Z}_+$ by the following equations:

$$x(k+1) = Ax(k) + B_u u(k) + B_d d(k) + A_\tau x(k-\tau) + B_{u\tau} u(k-\tau) + B_{d\tau} d(k-\tau), \quad (1a)$$

$$y(k) = Cx(k) + D_u u(k) + D_d d(k) + C_\tau x(k-\tau) + D_{u\tau} u(k-\tau) + D_{d\tau} d(k-\tau), \quad (1b)$$

where $x(k) \in \mathbb{R}^{n_x}$ are the water volumes, $u(k) \in \mathbb{R}^{n_u}$ represents the manipulated inputs, $y(k) \in \mathbb{R}^{n_y}$ denotes the outputs of the system and $d(k) \in \mathbb{R}^{n_d}$ are the lock operations, the flow from the PS and secondary canals, which can be regarded as disturbances. Moreover, $u(k-\tau)$ and $d(k-\tau)$ describe the delayed effect of the control actions and disturbances, respectively. Furthermore, $A, A_\tau, B_u, B_{u\tau}, B_d, B_{d\tau}, C, C_\tau, D_u, D_{u\tau}, D_d$ and $D_{d\tau}$ are time-invariant matrices of suitable dimensions.

The Calais canal is equipped with hydraulic devices whose operating range is constrained, and the water levels must be kept within an appropriate interval. Therefore, the constraints on the system variables in (1) are defined as follows:

$$\underline{u} \leq u(k) \leq \overline{u}, \quad (2a)$$

$$\underline{y}_{ref} - \alpha(k) \leq y(k) \leq \overline{y}_{ref} + \alpha(k), \quad (2b)$$

$$\underline{\alpha} \leq \alpha(k) \leq \overline{\alpha}, \quad (2c)$$

where $\overline{u}$ and $\underline{u}$ are the upper and lower operating limits of the actuators; $\overline{y}_{ref}$ and $\underline{y}_{ref}$ denote the upper (HNL) and lower (LNL) bounds of the normal navigation level (NNL) values, respectively. Besides, $\alpha(k)$ is a relaxation variable that is bounded with $\overline{\alpha}$ and $\underline{\alpha}$, to consider the flooding limits.

3.2 Moving horizon estimation formulation

The system states $x(k)$ are not directly measurable. Therefore, an MHE approach is designed to estimate the values of these states by solving a quadratic program using a moving estimation window of constant size (Rao et al., 2001). The moving window of size N_e contains only the most recent inputs and outputs. The formulation of the MHE is given below:

$$\min_{\hat{\mathbf{x}}_k} W^\top(k-N_e+1|k) P^{-1} W(k-N_e+1|k) + \sum_{i=k-N_e+1}^{k} \Big(W^\top(i|k) Q^{-1} W(i|k) + V^\top(i|k) R^{-1} V(i|k) \Big), \quad (3a)$$

subject, for $i \in \{k-N_e+1, ..., k\}$, to:

$$W(k-N_e+1|k) = \hat{x}(k-N_e+1|k) - x(k-N_e+1) \quad (3b)$$

$$W(i|k) = \hat{x}(i+1|k) - \Big(A\hat{x}(i|k) + B_u u(i|k) + B_d d(i|k) + B_{u\tau} u(i-\tau|k) + B_{d\tau} d(i-\tau|k) \Big), \quad (3c)$$

$$V(i|k) = y(i|k) - \Big(C\hat{x}(i|k) + D_u u(i|k) + D_d d(i|k) + D_{u\tau} u(i-\tau|k) + D_{d\tau} d(i-\tau|k) \Big), \quad (3d)$$

$$y(i|k) = y(i), \quad (3e)$$

$$\underline{x} \leq \hat{x}(j|k) \leq \overline{x}, j \in \{k-N_e+1, ..., k+1\} \quad (3f)$$

$$\hat{x}(l|k) = \hat{x}_{MHE}(l|k),\ l \in \{k-N_e-\tau+1, ..., k-N_e\}, \quad (3g)$$

$$u(m|k) = u_{db}(m|k),\ m \in \{k-N_e-\tau+1, ..., k\}, \quad (3h)$$

where R^{-1} and Q^{-1} are the weighting inverse matrices of appropriate dimensions specifying the confidence in the measurements and quality of the model, respectively, and $W(k-N_e+1|k)$ in (3b) is the error between $x(k-N_e+1)$, i.e., the most probable initial state, and $\hat{x}(k-N_e+1|k)$, i.e., the first value of the optimal state sequence, estimated by the MHE at time instant k. This error is weighted by matrix P^{-1}, which indicates the confidence into the initial state, and its tuning allows to satisfy estimation boundedness (Copp and Hespanha, 2017). Moreover, $y(i)$ are the measured water levels and u_{db} are the controls applied to the system, which are recorded in the database of managers. It is worth stressing again that the values u_{db} can differ from the optimal solution, as managers have the last word on the control actions that are applied.

The optimal sequence $\hat{\mathbf{x}}^*(k) = \{\hat{x}(j|k)\}_{j \in \mathbb{Z}_{[k-N_e+1,k+1]}}$ is determined by solving (3), and only the last value of the sequence $\hat{x}(k+1|k)$ is retained. Then, the problem is solved at the next time instant by shifting the moving window forward in time, i.e., exploiting the most recent information.

3.3 Model predictive control strategy

MPC is a well suited control approach for the management of canals, as it is characterized by predictive capabilities and determines the optimal control law (Camacho and Bordons, 2013). Therefore, a prediction model, given by the control-oriented model (1), is used to determine optimal control actions that minimize a certain criterion, e.g., multi-objective function, by solving an optimization problem over a prediction horizon N_p considering a set of physical and operational constraints. With all this, the MPC formulation reads as follows:

$$\min_{u_k, y_k, \alpha_k} \sum_{i=k}^{k+N_p-1} \Big(\ell^y(i|k) + \ell^e(i|k) + \ell^{\Delta u}(i|k) + \ell^\alpha(i|k) \Big), \quad (4a)$$

subject to, and for $i \in \{k, ..., k+N_p-1\}$:

$$x(i+1|k) = Ax(i|k) + B_u u(i|k) + B_d d(i|k) + B_{u\tau} u(i-\tau|k) + B_{d\tau} d(i-\tau|k), \quad (4b)$$

$$y(i|k) = Cx(i|k) + D_u u(i|k) + D_d d(i|k) + D_{u\tau} u(i-\tau|k) + D_{d\tau} d(i-\tau|k), \quad (4c)$$

$$\underline{y} - \alpha(i|k) \leq y(i|k) \leq \overline{y} + \alpha(i|k), \quad (4d)$$

$$\underline{u} \leq u(i|k) \leq \overline{u}, \quad (4e)$$

$$\underline{\alpha} \leq \alpha(i|k) \leq \overline{\alpha}, \quad (4f)$$

$$u(l|k) = u_{db}(l|k),\ l \in \{k-\tau, ..., k-1\}, \quad (4g)$$

where $k \in \mathbb{Z}_{\geq 0}$ is the current time instant, $i \in \mathbb{Z}_{\geq 0}$ is the time instant during the prediction horizon, $(k+i|k)$ indicates the predicted value of the variable at instant $k+i$ using information available at instant k, and $j \in \mathbb{Z}_{\geq 0}$ and $l \in \mathbb{Z}_{\geq 0}$ indicate the use of past actions applied to the system and stored in the manager's database (u_{db}) and information computed by the MHE ($\hat{x}_{MHE}$), respectively.

The different operational objectives included in (4a) are defined as follows:

- Keep water levels close to the navigation objective NNL:
$$\ell^y(k) \triangleq (y(k) - y_{ref})^\top W_y (y(k) - y_{ref}), \quad (5)$$
where y_{ref} is the vector of NNL values and W_y is a diagonal positive definite matrix.
- Minimize cost of operating equipment:
$$\ell^e(k) \triangleq u^\top(k) W_e u(k), \quad (6)$$

where W_e is a diagonal positive definite matrix, whose entries linked to pumping actions are much larger than those linked to the gates.

- Consider smoothness of control actions:

$$\ell^{\Delta u}(k) \triangleq \Delta u^\top(k) W_{\Delta u} \Delta u(k), \tag{7}$$

where $\Delta u(k) \triangleq u(k) - u(k-1)$, and $W_{\Delta u}$ is a diagonal positive definite matrix. The control actions should be smooth, i.e., Δu should be minimized, to extend the lifespan of component.

- Penalize relaxation of navigability condition:

$$\ell^{\alpha}(k) \triangleq \alpha^\top(k) W_{\alpha} \alpha(k), \tag{8}$$

where W_α is a diagonal positive definite matrix. The relaxation is useful to obtain optimal control actions even when the navigation conditions cannot be satisfied, but this situation is penalized.

Resolution of the MPC at time instant k yields the optimal sequences $\mathbf{u}^*(k) = \{u(i|k)\}_{i\in\mathbb{Z}_{[k,k+N_p-1]}}$, $\mathbf{y}^*(k) = \{y(i|k)\}_{i\in\mathbb{Z}_{[k,k+N_p-1]}}$ and $\boldsymbol{\alpha}^*(k) = \{\alpha(i|k)\}_{i\in\mathbb{Z}_{[k,k+N_p-1]}}$. However, only the first value $u_{MPC}(k) = u^*(k|k)$ from the sequence of optimal inputs is provided to the managers as optimal setpoint, and a new problem is solved at the next time instant.

3.4 Hierarchical control architecture

The hierarchical MPC proposed by Segovia et al. (2020) consists of three layers.

- The upper layer specifies the available hydraulic components according to the discrete state of the system. In low tide, pumps and gates are available; in high tide, the gates cannot be utilized. The manipulated inputs u_k are modified according to the tide.
- The intermediate layer is concerned with the resolution of the MHE and MPC. While the same MHE can be used for both tidal periods, a different MPC must be designed for each mode to account for the available hydraulic components.
- The lower layer is tasked with determining feasible flows that can be delivered by the hydraulic devices, and which are as close to the optimal control references (computed at the intermediate layer) as possible. In this work, a lookup table connecting flows and gate openings is proposed to determine the actions to be applied by managers.

4. HIERARCHICAL CONTROL STRATEGY TRANSCODING INTO PYTHON

The hierarchical control strategy presented in the previous section is designed using Matlab, and later transcoded into Python 3.9 [6]. The selected Integrated Development Environment is Pycharm [7], and the solver Gurobi [8] must be installed. Additional required libraries are given below; pip install {scipy, numpy, pandas, xlrd, cvxpy}. Using this tool and libraries, all layers and problem have been coded as functions, following the architecture depicted in Figure 4. The database query collects the measurements from the database using a sampling time $T_s = 2$ minutes. The data are summarized in Table 1, with binary variables at PS, and measurement values for levels.

[6] https://www.python.org/downloads/release/python-390/
[7] https://www.jetbrains.com/fr-fr/pycharm/
[8] https://www.gurobi.com/

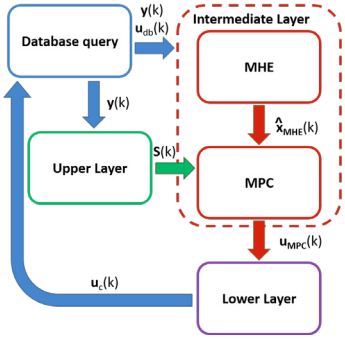

Fig. 4. Architecture of the Python code

Designation	Tag	Description
PS 3 Cornets	CA_3C_P1_MA	PUMP 1
	CA_3C_V1_MA	SCREW 1
	CA_3C_V2_MA	SCREW 2
PS Les Attaques	CA_LA_V1_MA	SCREW 1
	CA_LA_V2_MA	SCREW 2
PS Balinghem	CA_BA_P1_MA	PUMP 1
	CA_BA_P2_MA	PUMP 2
PS Nouvelle Eglise	CA_NE_V1_MA	SCREW 1
	CA_NE_V2_MA	SCREW 2
Level Les Attaques	CA_LA_NIV_PV	MEASUREMENT
Level Hennuin	VN_HE_NIV_PV	MEASUREMENT
Level Calais	CA_NIV_AM_PV	MEASUREMENT
Level Sea	CA_NIV_AV_PV	MEASUREMENT

Table 1. Required measurements from the database of French managers

From the measurements $\mathbf{y}(k)$, the levels of the canal in Calais and of the sea are provided to the upper layer. The sea level is compared to the canal level in Calais at each step time k. If the sea level is inferior to the canal level during three consecutive step times, the state of the system, denoted with $S(k)$, is low tide; otherwise it is high tide. The measurements $\mathbf{y}(k)$ and the control actions sent to the hydraulic devices $\mathbf{u}_{db}(k)$ are used by the MHE at the intermediate layer to estimate the state of the system $\hat{x}_{MHE}(k)$ every T_s time units, with $N_e = 12$ hours. The MPC function is executed with a control period $T_c = 2$ hours by considering $N_p = 12$ hours (covering 2 types of tide), but its value can be easily modified. Then, and based on the system state $S(k)$ and the estimated state $\hat{x}_{MHE}(k)$, the MPC is solved, providing control setpoint $u_{MPC}(k)$ every T_c time units.

The control setpoint $u_{MPC}(k)$ provides binary (On/Off) control setpoints for the pumps, and a certain flow (m^3/s) for gate G_1. The low layer aims at determining the control based on the Table 2, which determines the corresponding opening of the gate $u_c(k)$ according to the desired flow. This table is provided by the managers. The mean value of the discharge depends on the level in the canal and the sea level. This difference in terms of level and the time of gate opening depend on the type of tide. Three types can be considered according to the tidal coefficient C_{tide}: *spring tide* for $C_{tide} \geq 90$, *neap tide* for $C_{tide} \leq 45$, and *mean tide* for $45 < C_{tide} < 90$.

Finally, the control setpoints for the gate and pumps are provided to the managers every T_c time units. The corresponding values are sent to the database using the following tags:

- Setpoint for Gate: CA_G1_SP,
- Setpoint for Pump: CA_PUMP_SP.

Spring Tide	Neap Tide	Mean Tide	Gate
Mean flow	Mean flow	Mean flow	G_1 [dm]
0	0	0	0
1,9	1,4	1,7	1
2,7	2,2	2,4	2
4,2	3,4	3,8	3
6,6	5,4	6	4
7,6	6,3	7	5
8,6	7	7,8	6
9,4	7,9	8,7	7
11,1	8,9	10	8
11,6	9,4	10,5	10
12,3	9,8	11,1	11
12,7	10,3	11,5	12
13,2	10,6	11,9	13
13,9	11,1	12,5	14
14	11,3	12,7	15
14,3	11,5	12,9	16
14,5	11,7	13,1	17
14,7	12	13,4	18
14,8	12,2	13,5	20
14,9	12,3	13,6	22
15,1	12,4	13,7	24
15,2	12,5	13,8	25

Table 2. Gate opening G_1 as a function of mean flow [m^3/s], and according to the type of tide

An interface has been created to provide the optimal control setpoints $u_c(k)$) to the managers and tune the MHE and MPC parameters, and is depicted in Figure 5. Default values for weights in low/high tides are given. However, the managers can modify these values if the performance of the control strategy is not as expected. The horizon time can also be modified.

Fig. 5. Human Machine Interface for the predictive control architecture.

The complete Python code cannot be shown in this paper due to lack of space and relevance. However, the MHE code is given below to illustrate the approach:

```
# ---------------------- MHE ----------------------
def MHE_N(Ne, y_in, yf_in, u_in, wf_in, xo, x_prev, T_Sim, distf):

    # def MHE(y_in, yf_in, u_in, wf_in, xo, x_prev):
    coef_red = 0.001
    P = coef_red * ((xmax[0] ** 2) * sel2)
    Q = coef_red * ((xmax[0] ** 2) * sel2)
    R = coef_red * ((y_HNL[0] ** 2) * sel2)
    w_in = wf_in[:, n - 1:2 * n]

    x_est = Variable((size_Af, Ne + n + 1+1))

    obj_MHE = (quad_form((x_est[:, n + 1] - xo),  \
    np.linalg.inv(P)))
    constraints = []
    T_Sim=T_Sim

    for i in range(1, Ne+n+1):
        if i == n:
            obj_MHE +=(quad_form((x_est[:, i + 1] - \
                (Af @ x_est[:, i] + Bf @ w_in[:, i] + \
                Bdf @ distf[:, i + n + T_Sim - Ne] + \
                sel1 @ (Bf_n @ wf_in[:, i] + \
                Bdf_n @ distf[:, i + T_Sim - Ne]) + \
                sel2 @ (Bf_n @ w_in[:, 1] + \
                Bdf_n @ distf[:, i + T_Sim - Ne + 1]))),\
                np.linalg.inv(Q))) + \
                (quad_form((yf_in[:, 5] - \
                (Cf @ x_est[:, i] + Df @ w_in[:, i] + \
                Ddf @ distf[:, i + n + T_Sim - Ne] + \
                sel1 @ (Df_n @ wf_in[:, i] + \
                Ddf_n @ distf[:, i + T_Sim - Ne]) + \
                sel2 @ (Df_n @ w_in[:, 1] + \
                Ddf_n @ distf[:, i + T_Sim - Ne + 1]))),\
                np.linalg.inv(R)))

        elif i < 2 * n and i>n:
            obj_MHE += (quad_form((x_est[:, i + 1] - \
                (Af @ x_est[:, i] + Bf @ u_in[:, i - n] +  \
                Bdf @ distf[:, i + n + T_Sim - Ne] + \
                sel1 @ (Bf_n @ w_in[:, i - n] + \
                Bdf_n @ distf[:, i + T_Sim - Ne]) + \
                sel2 @ (Bf_n @ w_in[:, i - n + 1] + \
                Bdf_n @ distf[:, i + T_Sim - Ne + 1]))),\
                np.linalg.inv(Q))) + \
                (quad_form((y_in[:, i - n] - \
                (Cf @ x_est[:, i] + Df @ u_in[:, i - n] + \
                Ddf @ distf[:, i + n + T_Sim - Ne] + \
                sel1 @ (Df_n @ w_in[:, i - n] + \
                Ddf_n @ distf[:, i + T_Sim - Ne]) + \
                sel2 @ (Df_n @ w_in[:, i - n + 1] + \
                Ddf_n @ distf[:, i + T_Sim - Ne + 1]))),\
                np.linalg.inv(R)))

        elif i == 2 * n:
            obj_MHE += (quad_form( (x_est[:, i + 1] - \
                (Af @ x_est[:, i] + Bf @ u_in[:, i - n] + \
                Bdf @ distf[:, i + n + T_Sim - Ne] + \
                sel1 @ (Bf_n @ w_in[:, i - n] + \
                Bdf_n @ distf[:, i + T_Sim - Ne]) + \
                sel2 @ (Bf_n @ u_in[:, 1] + \
                Bdf_n @ distf[:, i + T_Sim - Ne + 1]))),\
                np.linalg.inv(Q))) + \
                (quad_form((y_in[:, i - n] - \
                (Cf @ x_est[:, i] + Df @ u_in[:, i - n] + \
                Ddf @ distf[:, i + n + T_Sim - Ne] + \
                sel1 @ (Df_n @ w_in[:, i - n] + \
                Ddf_n @ distf[:, i + T_Sim - Ne]) + \
                sel2 @ (Df_n @ u_in[:, 1] + \
                Ddf_n @ distf[:, i + T_Sim - Ne + 1]))),\
                np.linalg.inv(R)))

        elif i> 2*n:
            obj_MHE += (quad_form((x_est[:, i + 1] - \
                (Af @ x_est[:, i] + Bf @ u_in[:, i - n] + \
                Bdf @ distf[:, i + n + T_Sim - Ne] + \
                sel1 @ (Bf_n @ u_in[:, i - 2 * n] + \
                Bdf_n @ distf[:, i + T_Sim - Ne]) + \
                sel2 @ (Bf_n @ u_in[:, i - 2 * n + 1] + \
                Bdf_n @ distf[:, i + T_Sim - Ne + 1]))),\
                np.linalg.inv(Q))) + \
                (quad_form((y_in[:, i - n] - \
                (Cf @ x_est[:, i] + Df @ u_in[:, i - n] +\
                Ddf @ distf[:, i + n + T_Sim - Ne] + \
                sel1 @ (Df_n @ u_in[:, i - 2 * n] + \
                Ddf_n @ distf[:, i + T_Sim - Ne]) + \
                sel2 @ (Df_n @ u_in[:, i - 2 * n + 1] + \
                Ddf_n @ distf[:,i + T_Sim - Ne + 1]))),\
                np.linalg.inv(R)))

    constraints += [x_est >= np.matlib.repmat(\
    np.reshape(xmin, (2, 1)), 1, Ne + n + 1+1)]
    constraints += [x_est <= np.matlib.repmat(\
    np.reshape(xmax, (2, 1)), 1, Ne + n + 1+1)]
        # constraints += [x_est[:,i] >= xmin]
```

```
        # constraints += [x_est[:,i] <= xmax]
constraints += [x_est[:, n + Ne] == x_prev]

prob = Problem(Minimize(obj_MHE), constraints)
prob.solve(solver=GUROBI, warm_start=True)

x_est_out = x_est[:, Ne + n + 1].value
return x_est_out
```

5. CONCLUSION

In this paper, the transcoding of a hierarchical MPC strategy for the Calais canal into Python is proposed. The original control strategy is based on MHE and MPC, and was designed using Matlab. Transcoding is carried out with the objective to implement this solution within the information system of the managers, which allows to test the designed predictive control strategy on the real system. Next steps regard its implementation on the information system of the managers to obtain feedback (which will be used to improve the solution) and gain insight on its performance. In particular, performance obtained during extreme weather situations, e.g., flooding periods, is of great relevance, and will lead to future research on strategies to mitigate their effects.

ACKNOWLEDGEMENTS

The authors want to thank the *Institution Intercommunale des Wateringues* for providing data and information regarding the management of the Calais canal.

REFERENCES

Baunsgaard, K.M.H., Ravn, O., Kallesøe, C.S., and Poulsen, N.K. (2017). Mpc control of water supply networks. In *2016 European Control Conference, ECC 2016*, 1770–1775.

Camacho, E.F. and Bordons, C. (2013). *Model predictive control*. Springer science & business media.

Chen, W.., Shang, C., Zhu, S., Haldeman, K., Santiago, M., Stroock, A.D., and You, F. (2021). Data-driven robust model predictive control framework for stem water potential regulation and irrigation in water management. *Control Engineering Practice*, 113.

Copp, D.A. and Hespanha, J.P. (2017). Simultaneous nonlinear model predictive control and state estimation. *Automatica*, 77, 143–154.

Deshays, R., Segovia, P., and Duviella, E. (2021). Design of a MATLAB HEC-RAS Interface to Test Advanced Control Strategies on Water Systems. *Water*, 13(6).

Duviella, E. and Hadid, B. (2019). Simulation tool of the calais canal implementing logic control based regulation. *CMWRS, Delft, The Netherlands, 19-20 September*.

Gehbauer, C. (2019). Framework for multi layer control in python (fmlc) v1.0. [Computer Software] https://doi.org/10.11578/dc.20191029.5.

Grosso, J.M., Velarde, P., Ocampo-Martinez, C., Maestre, J.M., and Puig, V. (2017). Stochastic model predictive control approaches applied to drinking water networks. *Optimal Control Applications and Methods*, 38(4), 541–558.

Guekam, P., Segovia, P., Etienne, L., and Duviella, E. (2021). Hierarchical model predictive control and moving horizon estimation for open-channel systems with multiple time delays. In *2021 European Control Conference (ECC)*, 198–203.

Guo, C. and You, F. (2019). A data-driven real-time irrigation control method based on model predictive control. In *Proceedings of the IEEE Conference on Decision and Control*, volume 2018-December, 2599–2604.

Horváth, K., Rajaoarisoa, L., Duviella, E., Blesa, J., Petreczky, M., and Chuquet, K. (2015). *Enhancing Inland Navigation by Model Predictive Control of Water Levels: The Cuinchy-Fontinettes Case*, 211–234. Springer International Publishing, Cham.

Karimi Pour, F. and Puig, V. (2021). Health-aware model predictive control including fault-tolerant capabilities for drinking water transport networks. In *2021 29th Mediterranean Conference on Control and Automation, MED 2021*, 663–668.

Karimi Pour, F., Puig, V., and Cembrano, G. (2018). Health-aware lpv-mpc based on system reliability assessment for drinking water networks. In *2018 IEEE Conference on Control Technology and Applications, CCTA 2018*, 187–192.

Nguyen, L., Prodan, I., Lefevre, L., and Genon-Catalot, D. (2017). Distributed model predictive control of irrigation systems using cooperative controllers. In *IFAC-PapersOnLine*, volume 50, 6564–6569.

Ranjbar, R., Duviella, E., Etienne, L., and Maestre, J.M. (2020). Framework for a digital twin of the canal of calais. *Procedia Computer Science*, 178, 27–37. doi: https://doi.org/10.1016/j.procs.2020.11.004. 9th International Young Scientists Conference in Computational Science, YSC2020, 05-12 September 2020.

Rao, C.V., Rawlings, J.B., and Lee, J.H. (2001). Constrained linear state estimation—a moving horizon approach. *Automatica*, 37(10), 1619–1628.

Segovia, P., Duviella, E., and Puig, V. (2020). Multi-layer model predictive control of inland waterways with continuous and discrete actuators. *IFAC-PapersOnLine*, 53(2), 16624–16629. 21st IFAC World Congress.

Segovia, P., Puig, V., Duviella, E., and Etienne, L. (2021). Distributed model predictive control using optimality condition decomposition and community detection. *Journal of Process Control*, 99, 54–68.

Segovia, P., Rajaoarisoa, L., Nejjari, F., Puig, V., and Duviella, E. (2017). Decentralized control of inland navigation networks with distributaries: application to navigation canals in the north of France. *ACC'17, Seattle, WA, USA, May 24-26*.

Segovia, P., Rajaoarisoa, L., Nejjari, F., Duviella, E., and Puig, V. (2019). Model predictive control and moving horizon estimation for water level regulation in inland waterways. *Journal of Process Control*, 76, 1–14.

Sheik, A.G., Tejaswini, E., Seepana, M.M., Ambati, S.R., Meneses, M., and Vilanova, R. (2021). Design of feedback control strategies in a plant-wide wastewater treatment plant for simultaneous evaluation of economics, energy usage, and removal of nutrients. *Energies*, 14(19).

Sun, C., Romero, L., Joseph-Duran, B., Meseguer, J., Palma, R.G., Puentes, M.M., Puig, V., and Cembrano, G. (2021). Control-oriented quality modelling approach of sewer networks. *Journal of environmental management*, 294.

Svensen, J.L., Sun, C., Cembrano, G., and Puig, V. (2021). Chance-constrained stochastic mpc of astlingen urban drainage benchmark network. *Control Engineering Practice*, 115.

Takács, B., Holaza, J., Števek, J., and Kvasnica, M. (2015). Export of explicit model predictive control to python. In *2015 International Conference on Process Control*, 78–83.

Zheng, Z., Wang, Z., Zhao, J., and Zheng, H. (2019). Constrained model predictive control algorithm for cascaded irrigation canals. *Journal of Irrigation and Drainage Engineering*, 145(6).