The open real-time control (RTC)-Tools software framework for modeling RTC in water resources systems
D. Schwanenberg, B. P. J. Becker and M. Xu

ABSTRACT

Real-time control-Tools is a novel software framework for modeling real-time control and decision support in water resources systems. It integrates different control paradigms ranging from simple feedback control strategies with triggers, operating rules and controllers to advanced optimization-based approaches such as model predictive control (MPC). A key feature of the package is the modular integration of modeling components, related adjoint models, and optimization algorithms which makes it well suited for the control of large-scale water systems. Interfaces enable its integration into Supervisory Control and Data Acquisition systems, operational stream flow forecasting, and decision support systems as well as hydraulic modeling packages. This paper presents an overview of the novel software framework, gives an introduction into the underlying control theory for which it has been developed and discusses the related software architecture. A first case describes an innovative combination of binary decision trees and feedback control in application to the modeling of a highly regulated River Rhine reach along the German–French border. Two additional cases present the efficient application of MPC to the short-term management of two large-scale water systems in the Netherlands and the USA.

Key words | feedback control, model predictive control, real-time control, short-term reservoir management

INTRODUCTION

Overview

Past and current hydraulic engineering activities have had significant impacts on many river basins around the world. Examples cover multi-purpose storage reservoirs for flood mitigation, water supply, irrigation, hydroelectricity, or recreation as well as river training measures including the construction of weirs and run-of-river hydro power projects to enable navigation and generate electricity. In low-lying delta countries such as the Netherlands, engineering measures over many centuries led to man-made water systems to drain the land, defend the country against floods, and supply inhabitants, industry, and agriculture with freshwater (van Steen & Pellenbarg 2004).

The main motivation for implementing engineering measures in water resources systems is to gain control over the hydrologic cycle. On the one hand, this refers to defensive structural measures for flood protection such as river dikes, which act by their presence, but do not have controllable components. These will be of minor interest to us. On the other hand, hydraulic structures such as segment weirs, pumps, and dams are actively operated by stakeholders in real-time and operation options are various. The outflow from a dam of a storage reservoir, for example, can be realized by a bottom outlet, a hydropower turbine, or a gated spillway, and it can be shifted in time related to the reservoir inflow by making strategic or tactical use of the reservoir storage capacity. The stakeholder’s intention is to operate hydraulic structures in such a way that they serve specific objectives in the best possible way. Finding these control trajectories is a non-trivial task and we will...
A straightforward approach for defining control actions is to make them dependent on available observations and a predefined operating procedure by feedback and feed-forward control (Simonović & Unesco 2009). An implicit feature of this approach is that we can only react to an event and take measures after it is observed. From a computational point of view, its implementation is straightforward and only requires simulation, which runs the control as well as related hydrologic and hydraulic models in an open (feed-forward) or closed loop (feedback). If the control strategy becomes more complex and includes conditional components for switching between different controllers, the main challenge is the proper handling of these conditions. We present an approach based on binary decision trees which is fail-safe by definition. To let control actions not only depend on current or historical states, but also on future system states, or, more precisely, expected future system states, is a logical advancement. We refer to this approach as predictive control. Model predictive control (MPC) enables us to anticipate future events and take actions before these events are observed (Zavala et al. 2009).

Feedback and feed-forward control actions must be predefined for all possible situations. The operational strategy (what to achieve) is usually translated into an operational protocol which defines how the water system is controlled. Such a protocol can contain what-if instructions or decision trees. When applying predictive control, the operational objectives (what to achieve) are usually translated into a mathematical optimization problem. The question how to operate the water system is determined within an optimization algorithm. From the modeling point of view, feedback and predictive control techniques require different technical set-ups. Feedback control works like an additional simulation component, which is coupled on a time-step basis to other hydrological modeling components.

In contrast, the MPC option uses external forecasts, also referred to as disturbances in control terminology, in combination with an internal representation of the controlled system (García et al. 1989). An optimization algorithm computes the optimal control trajectories over a finite forecast horizon. This approach is called an online approach and solves an optimization problem at forecast time in comparison to offline approaches which use optimization for the parametrization of feedback rules; see Labadie (2004) for a review of various optimization techniques for multi-reservoir systems. Solving optimization is computationally expensive and may restrict its real-time application, so dedicated prediction models of the controlled part of the water system are required. These prediction models are often simplified in order to make them fast enough for running within the optimization procedure. For an improved performance and in order to avoid truncation errors, the prediction model should not only compute system states, but also back-propagate the derivatives of the objective function and constraints to the control parameters through the model. This can be achieved for example by algorithmic differentiation (Griewank & Walther 2008). The last feature determines the Real-time control (RTC)-Tools software architecture to a large extent.

The following sections of the introduction provide the theoretical concepts of feedback and feed-forward control as well as MPC. Then we present and discuss the conceptual design of the RTC-Tools software framework and, in particular, introduce an innovative framework for the modular set-up of simulation models and its adjoints within optimization schemes. Furthermore, this paper presents a number of practical control applications. In the first case, the novel framework simulates complex feedback control strategy in combination with a one-dimensional feedback control (RTC) model of the highly regulated section of the Rhine River along the German–French border between the cities of Basel and Karlsruhe. In a second case, the MPC option is used to drain the polder system of the Dutch regional water authority Noorderzijlvest (NZV) in the north of the Netherlands. The last case demonstrates a large-scale application of MPC for the short-term optimization of the Federal Columbia River Power System (FCRPS), managed by the Bonneville Power Administration (BPA), for dispatching the hydro power generation under consideration of environmental and other operational constraints.

**Feedback control and feed-forward control**

Two frequently used techniques for operating hydraulic structures in water resources systems are feedback and feed-forward control. For the example of maintaining a
water level at target by adjusting the downstream gate position, applying feedback control means to use the control error, i.e., the difference between set point and actual water level at the gate, to determine the control action. If the water level is above the set point, the operator will increase the gate opening and vice versa. This action directly feeds back into the observed water level. In contrast, feed-forward control is a technique wherein an external disturbance, i.e., an observation that is independent of the controlled system, is used to determine the control action. The operator could decide to open the gate when it starts to rain in order to account for additional water flow.

An example of a widely used controller for hydraulic structures is the proportional-integral-derivative (PID) controller. It is a generic feedback controller with an optional feed-forward term according to

$$ e(t) = x(t) - x_{sp}(t) $$

$$ y(t) = K_p e(t) + K_i \int_0^t e(\tau) \, d\tau + K_d \frac{d}{dt} e(t) + K_d d(t) $$

(1)

where \( e(t) \) is the control error, i.e., the difference between a process variable \( x(t) \) and its set point \( x_{sp}(t) \) at time \( t \); \( K_p \), \( K_i \), and \( K_d \) are the proportional, the integral and the derivative gains, respectively. The integral term accounts for the control error in the period \( 0 \leq \tau \leq t \). The optional feed-forward term consists of a factor \( K_f \) and an external disturbance \( d(t) \). The setting of a structure \( y(t) \) is the controller output.

The discrete form of Equation (1) reads

$$ e^k = x^{k-1} - sp^{k-1} $$

$$ E^k = E^{k-1} + \Delta t e^k $$

$$ y^k = K_p \cdot e^k + K_i \cdot E^k + K_d \cdot \frac{e^k - e^{k-1}}{\Delta t} + K_d \cdot d^k $$

(2)

where \( E \) is the integral of \( e \) and \( \Delta t \) is the simulation time step.

Note the practical implementation of a PID controller includes additional computational steps for the optional limitation of the maximum velocity of the controller output or an up-wind correction to limit the controller’s integral part in order to increase its control performance. However, our main interest at this point is the fact that the controller output at time step \( k \) is exclusively computed based on data at time step \( k - 1 \) and \( k \) (in case of the external disturbance). This convention is common to all other feedback control techniques.

Beside a layer of controllers, which determines how a structure is operated on the level of the actuator, practical control strategies often include a decision layer for selecting a dedicated controller for specific operating conditions. In a cascade of run-of-river hydropower plants and weirs, we may want to keep the water levels sufficiently high under drought conditions in order to support cargo ship navigation, use some of the system flexibility under normal flow conditions for hydropower peaking, and appoint the control to flood mitigation in case of approaching flood events.

The simplest way to trigger between different operating conditions is to evaluate a relational condition of the type

$$ y^k = \begin{cases} 1 & \text{if } x^{k-1} > \theta \\ 0 & \text{otherwise} \end{cases} $$

(3)

where \( x \) is an arbitrary model state, \( \theta \) a user-defined threshold value, and \( y \) the trigger output, returning either true (1) or false (0). Beside the \( > \) operator, we may also use other relational operators such as \( >, \geq, =, \neq, < \), or \( < \). In RTC-Tools, we refer to Equation (3) as the ‘standard trigger’.

A more advanced trigger is the so-called ‘deadband trigger’. It checks the input for an upper or lower threshold crossing. The trigger is active in case of an up-crossing of the upper threshold \( \theta_u \) and it is inactive in case of a down-crossing of the lower threshold \( \theta_l \). In the range in-between, the trigger keeps its former state. It reads

$$ y^k = \begin{cases} 1 & \text{if } x^{k-1} \neq \theta_u \\ 0 & \text{if } x^{k-1} < \theta_l \\ y^{k-1} & \text{otherwise} \end{cases} $$

(4)

where \( x \) again is an arbitrary model state. Dead band triggers are often used to switch on and off pumps or turbines based on a water level observation. In this case, the dead band prevents wear by ensuring that the device is not switched on or off due to small variations in the water level only. This becomes important in those cases where the corresponding control operation has a direct impact on the system states \( x \),
for example, for a turbine that is switched on and off based on the water level upstream.

For more complex control strategies, the results of several triggers can be combined by logical conditions. We introduce a formulation based on binary decision trees such as given in Figure 1 for representing hierarchical conditions. This formulation leads to unique control actions by definition by only activating the controller on the unique active path of the decision tree and deactivating all others. Figure 1 presents an example of such a set-up for choosing the appropriate operating rule for a river weir.

**Model predictive control**

While feedback control means to operate a water system based on historical and current system states, MPC uses predicted future state trajectories, for example, for anticipating approaching flood events within current control actions. This requires internal modeling of the controlled water system to assess and optimize the impact of control actions on the controlled system. While feedback and feed-forward control actions are based on the current system state, MPC takes the future behavior of the water system into account.

In the mathematical formulation of the approach (Schwanenberg et al. 2012), we consider a discrete time dynamic system according to

\[
\begin{align*}
  x^k &= f(x^{k-1}, x^k, u^k, d^k) \\
  y^k &= g(x^k, u^k, d^k)
\end{align*}
\]

where \( x, y, u, d \) are respectively the state, dependent variable, control and disturbance vectors, and \( f() \) and \( g() \) are functions representing an arbitrary linear or nonlinear water resources model. If being applied in MPC, Equation (5) is used to predict future trajectories of the state \( x \) and dependent variable \( y \) over a finite time horizon represented by \( k = 1, \ldots, N \) time instants, in order to determine the optimal set of controlled variables \( u \) by an optimization algorithm. Under the hypothesis of knowing the realization of the disturbance \( d \) over the forecast horizon, a sequential MPC approach (Diehl et al. 2009), also referred to as Single Shooting, can be formulated as follows:

\[
\begin{align*}
  \min_{u \in \{0, \ldots, T\}} & \sum_{k=1}^{N} J(x^k(u, d), y^k, u^k) + E(x^N(u, d), y^N, u^N) \\
  \text{subject to} & \quad h(x^k(u, d), y^k, u^k, d^k) \leq 0, \quad k = 1, \ldots, N
\end{align*}
\]

![Figure 1](http://example.com/figure1.png) Binary decision tree to distinguish low flow, medium flow, and flood conditions with dedicated operating rules for keeping the upstream water level on set point, conducting hydro power peaking for increasing power production revenues and mitigating flood events, respectively.
where \( \hat{x}^k(u, d) \) and \( \hat{y}^k \) are simulation results, \( J() \) is a cost function associated with each state transition, \( E() \) is an additional cost function for the final state condition, and \( h() \) are hard constraints.

In the corresponding simultaneous or collocated optimization approach (Diehl et al. 2009), the states \( x \) and dependent variables \( y \) become additional variables of the optimization problem, and the process model (Equation (5)) gets an equality constraint of the optimization according to

\[
\min_{u, x, y \in [0..T]} \sum_{k=1}^{N} J(x^k, y^k, u^k) + E(x^N, y^N, u^N)
\]

subject to \( h(x^k, y^k, u^k, d^k) \leq 0 \), \( k = 1, \ldots, N \)

\( x^k - f(x^{k-1}, x^k, u^k, d^k) = 0 \)

\( y^k - g(x^k, u^k, d^k) = 0 \)

(7)

Both methods lead to identical solutions, but have pros and cons for specific optimization problems in terms of runtime performance. The sequential approach (Equation (6)) is more efficient, if hard constraints are defined on control variables \( u \) only and gets inefficient for hard constraints on states \( x \) or dependent variables \( y \). Because of the state dependency on all prior control variables \( u \), the constraint Jacobian becomes dense in the lower triangular matrix. In this case, the simultaneous approach (Equation (7)) shows better efficiency, since states are an (independent) input of the optimizer and the constraint Jacobian gets sparse (Xu & Schwanenberg 2012), although the dimension of the optimization problem and the number of constraints is significantly larger.

A mix of both methods leads to the so-called Multiple Shooting approach. It combines the advantages of both methods and uses the simultaneous set-up only for modeling components with hard constraints. For example, the optimization model of a reservoir and a downstream river reach with unregulated laterals can be modeled with a simultaneous set-up for the reservoir and a sequential one for the river reach downstream. By introducing the forebay elevation as an optimization variable and an equality constraint on the mass balance, hard constraints can efficiently be defined for the forebay elevation of the reservoir, i.e., minimum and maximum operating limits, and for dependent variables such as the tailwater elevation or the power generation. Since the downstream river reach is only partially controlled by the reservoir because of the contributing unregulated laterals, the definition of hard constraints on the states of the routing model, i.e., a maximum stage or discharge at an inundation area, may lead to an infeasible solution space of the optimization. In this case, soft constraints in combination with a sequential set-up are more appropriate. This keeps the dimension of the optimization problem small and contributes to an efficient solution as well.

THE RTC-TOOLS SOFTWARE FRAMEWORK

History

A forerunner of RTC-Tools originated in 2007 from the integration of several project-specific reservoir simulation modules for flood forecasting systems in river basins in Austria, Germany, and Pakistan. We refer to this reservoir model as the Delft-FEWS Reservoir Module (FEWS-RM), because the mentioned forecasting systems have in common that they are based on the forecasting framework Delft-FEWS (Werner et al. 2013).

FEWS-RM aimed at the simulation of pool routing in reservoirs including related operating rules with the option to overrule the simulated control by interactive user input from the forecasting system. Features for supporting a sequential nonlinear version of MPC were introduced in 2008 and extended beyond. This included the implementation of simplified hydraulic models such as the kinematic or diffusive wave models as internal model of the MPC as well as the introduction of related adjoint models for effectively computing first-order derivatives of the objective function.

In 2010, the consolidation of existing features of the FEWS-RM and new requirements regarding interfaces and more advanced control techniques led to a complete redesign of the software we now refer to as RTC-Tools. In this context, the original computational core in Java was restructured and implemented in C++ for a better integration with existing numerical libraries and to increase the computational performance. Furthermore, binary decision trees have been introduced with a suite of triggers to switch...
controllers and operating rules on and off to enable the simulation of more sophisticated feedback control strategies. Additional features are C# interfaces to OpenMI (Gregersen et al. 2007) and DeltaShell (Donchyts & Jagers 2010), the MATLAB plugin, and a better compliance with OpenDA (OpenDA Association 2013) for conducting automatic model calibration and data assimilation.

In 2011 and beyond, the main focus has been on the extension of the MPC options. This covers the implementation of a simultaneous MPC mode with the option to mix it with the existing sequential mode resulting in a Multiple Shooting scheme, and features for explicitly taking into account forecast uncertainty in the MPC (Raso et al. 2013). RTC-Tools was first released in late 2012.

License and availability

RTC-Tools is available under a dual license model. We primarily provide the software under the GNU General Public License, version 2.0 (GPL2) as open source software. The related open source site (http://oss.deltares.nl/web/rtc-tools) provides information about the software, its application, a download section for binaries, manuals as well as further information and a subversion repository of the source code. As an alternative for business critical applications of end users, we provide RTC-Tools under a proprietary license. This enables us to integrate the software with commercial third-party products, in particular for completing and replacing the open source solvers for linear equation systems and optimization problems by higher performance components.

Software architecture

Figure 2 presents an overview of the RTC-Tools software architecture. We distinguish four main components:

- The Time Series Database (TSD) module handles the access to time series data. It makes use of a three-dimensional array according to \( v(e, k, s) \) where \( v \) is an arbitrary model input, state, or output and the indices represent the ensemble member \( e \), the time step \( k \) and the time series id \( s \). Scalars are represented by a single index \( s = i \), one-dimensional vectors, and two-dimensional grids by an index range \( s = i \ldots i + n \). If using the algorithmic differentiation for computing first-order sensitivities, we keep a duplicate array for the adjoint variable \( v_a(e, k, s) \).

![Software architecture of RTC-Tools including file input/output and the computational core.](image-url)
• The Model Schematization Module (MSM) includes a model library with simulation components. These implement a simulation and first-order sensitivity mode each. The simulation mode propagates the model forward in time by computing new system states and model outputs $v^k$ by the functions $f()$, $g()$ in Equation (5) and the old state $v^{k-1}$. Note this includes also all feedback components as well as the decision tree approach described in the section Feedback control and feed-forward control, and enables these components to restart from a single state. It implies by definition that a continuous simulation and one we split up give identical results. This is an important feature in operational systems, for example, to update historical model states or in application to data assimilation by Kalman filters. In the first-order sensitivity mode, also referred to as the adjoint mode, adjoint variables at the old state $v^{k-1}_a$ are computed backwards in time by the adjoint models $f_a()$, $g_a()$ based on reverse algorithmic differentiation (Griewank & Walther 2008) and the adjoint variables at the new state $v^k_a$ as well as both system states at steps $k-1$ and $k$. The latter is the main motivation for the TSD to keep variables and its adjoints in-memory for all time steps over the simulation horizon. So far, all existing adjoint models of the library have been derived manually. Although labor-intensive, we found the effort acceptable since most modeling components used in the optimization are relatively simple. Furthermore, the manual set-up in comparison to automatic differentiation algorithms (Bell & Burke 2008) leads to more effective implementations for implicit equations.

• The Runtime Module (RM) includes the definition of the optimization problem as well as the application logic for conducting simulation, first-order sensitivity, and optimization runs. The simulation implements a loop over the ensembles, time steps, simulation components and evaluates the objective function terms and constraints. The additional first-order sensitivity mode computes the objective function gradient $df/dz$ of the objective function $f$ with respect to the optimization variables $z$ and the constraint Jacobian $dh/dz$. The total derivatives are achieved by initializing the adjoint matrix $v_a(k, s)$ with the partial derivatives $df/\partial v$, $dh/\partial v$ with respect to all variables $v$ and executing the time step and simulation component loop backwards in time and space with the selected components of the MSM. Note the relation $\mathbb{R}^n \rightarrow \mathbb{R}$ of the objective function $f(z)$, i.e., the computation of a single objective function value from $n$ optimization variables, motivates the use of backwards differentiation whereas a forward technique would be more computational effective for a relation $\mathbb{R} \rightarrow \mathbb{R}^n$ (Griewank & Walther 2008). It remains to interface the functions for computing objective function values, constraints, and related derivatives as well as some additional functions for the administration of the optimization problem to an optimization algorithm. This is either the embedded IPOPT 3.11 optimizer (Wächter & Biegler 2006) including the optional HSL 2013 linear equation solvers or an external optimizer linked by one of the in-memory interfaces described in the next section.

• The File Input/Output Module (FIOM) takes care of the parsing and serialization from and to XML files. The parser code is automatically generated from the XSD schema definition based on the data binding software Code Synthesis XSD 3.3.0 and the Xerces C++ 3.1.1 XML parsing library. The procedure reduces the integration effort of new components significantly and leads to a maximum of consistency between file inputs and the software code. The XSD schemas are furthermore used to document the set-up options and validate the XML configuration carried out by the users.

An exception from the procedure above is the I/O of time series data. Whereas the use of pure XML is acceptable for smaller models, it becomes a computational burden for models with a large amount of time series data due to the overhead in the XML files. Therefore, the FIOM features an option to keep only meta information in XML and supply mass data in a binary file.

A final overall topic of the framework is parallelization. It is primarily implemented on the level of the ensemble loop of the RM by assigning the execution of simulation or first-order sensitivity executions to parallel threads. In this context, the MSM is read-only from a thread-perspective and receives dynamical data as function argument to propagate the simulation or sensitivity mode. Further parallelization is applied within the optimization algorithms, in particular on the level of the linear equation solvers and libraries with Basic Linear Algebra Subprograms.
Interfaces

RTC-Tools is primarily used in combination with or embedded in other software packages. Thus, interfaces and adapters constitute a key ingredient of the software. This section will give an overview of these interfaces and typical application.

From the very beginning of the development, RTC-Tools has been used within the forecasting system Delft-FEWS (Werner et al. 2013). The Delft-FEWS PI-XML file format of the Delft-FEWS general adapter is used in RTC-Tools as the native format for reading and write time series, storing states, externalizing parameters, or writing diagnostics. The interaction between the two components is triggered by Delft-FEWS according to the following sequence:

1. Time series data and other model inputs are prepared by Delft-FEWS in the PI-XML file format.
2. Delft-FEWS calls the RTC-Tools executable, which reads the input, executes the model, and writes output back into PI-XML.
3. Delft-FEWS imports the RTC-Tools outputs.

RTC-Tools models are currently being used in about 15 operational Delft-FEWS systems around the world. Main fields of application include the operational decision-making for reservoirs and the control of drainage or polder systems. Furthermore, internal RTC-Tools models have been used to complete model chains in general forecasting applications, for example, by supplying simulation components for reservoirs and their operating rules. Note that the Delft-FEWS general adapter facilitates data exchange via files prior to the simulation and not during runtime like the interfaces we describe in the following.

Important features for prototyping MPC applications are the interfaces to the high-level programming environments MATLAB and the General Algebraic Modeling System (GAMS). The MATLAB interface was implemented at an early stage of the development and has been broadly used and extended over the years. It enables the user to get and set time series data, set external parameters, and replace the embedded IPOPT optimizer by external ones in MATLAB by exchanging the objective function value, constraints, and their derivatives. We frequently use the interface for the pre-processing and visualization of data, the automatic calibration of internal models, or the prototyping of novel MPC features. Through MATLAB and extensions such as TOMLAB, the MPC can be extended to mixed-integer formulation to consider binary or integer inputs and logical constraints. A novel and relatively new interface connects RTC-Tools to GAMS via its ‘external function’ feature. This enables the integration of RTC-Tools models with internal GAMS models.

For water systems with complex operational protocols, it can make sense to model hydrodynamic processes and real-time control separately in order to choose the appropriate component independently for each domain. In this set-up, the RTC-Tools model represents human operations or automated control of the water system, the hydrodynamic model, e.g., a one-dimensional open channel flow model, represents hydrodynamics and hydraulic structures, and both models exchange data in-memory in each simulation time step. A case for such a set-up is described below in the section Hydraulic model with feedback control of the Upper Rhine River. For conjunctive modeling of real-time control and hydrodynamics, RTC-Tools provides the open modeling interface OpenMI (Gregersen et al. 2007). Within an OpenMI composition RTC-Tools and hydrodynamic model data are exchanged explicitly (Morita & Yen 2000). This means a control action in the current time step \( k \) affects the next time step \( k + 1 \) of the hydrodynamic model. This coupling method is also called ‘time-lagged approach’ (Fairbanks et al. 2001).

The integrated modeling framework DeltaShell (Donchyts & Jagers 2010) facilitates data exchange between different model components at runtime, but also provides a graphical user interface (GUI) for the model building process and the system analysis. The feedback control part of RTC-Tools is integrated in DeltaShell as D-RTC and forms part of version 3 of the hydraulic modeling package SOBEK. Figure 3 presents an operational protocol with a decision tree, triggers, and operating rules in D-RTC under DeltaShell. Triggers and operating rules are editable, input and output data from a coupled hydrodynamic model or water quality model is visualized in the decision tree and in maps. Model results of a conjunctive simulation can be analyzed and displayed coherently.
APPLICATIONS

Hydraulic model with feedback control of the Upper Rhine River

The River Rhine reach between the cities Basel and Karlsruhe is the most upstream part of the Upper Rhine, a 362 km long Rhine section between Basel and Bingen. Its southern part forms the border between Germany and France. Between the years 1928 (start of construction of Grand Canal d’Alsace) and 1977 (finalization of barrage Iffezheim), the Upper Rhine was heavily modified in order to make it navigable and to generate hydropower. The modifications have been realized in three different ways:

2. Construction of weirs with bypasses between the cities Breisach and Strasbourg.
3. Complete canalization between the cities Strasbourg and Iffezheim.

Figure 4 shows a schematic overview of the hydraulic system of the Upper Rhine with the three different types of river training measures. Between Basel and Breisach, the discharge is divided between the Canal d’Alsace and the old river bed. Up to its design discharge of 1,400 m³/s, water is primarily diverted into the canal in order to provide the four hydropower stations at Kembs, Ottmarsheim, Fessenheim, and Vogelgrün with water. Weir Kembs is controlled such that a minimum discharge of 20 m³/s in winter and 30 m³/s in summer is maintained in the old river bed. Weir Breisach has been built in the old branch for stabilizing
the groundwater table, which decreased severely after the construction of the Grand Canal d’Alsace. The old river branch upstream of Weir Breisach is also used for flood retention during flood periods.

Several weirs are installed between Breisach and Strasbourg. Four bypass canals enable ships to circumvent these weirs. At the downstream end of each bypass, a hydropower plant and a lock are installed. This approach, which was completed between the years 1956 and 1970, was chosen to leave the water levels in the old river branch almost unchanged. The lateral canals have a discharge of up to 1,400 m³/s. A minimum discharge of 15 m³/s is maintained in the old river branch. The weir Kehl/Strasbourg has been built for stabilizing the groundwater level along the barrage section Strasbourg. It is also used for flood retention purposes. In the river section between Strasbourg and Iffezheim, the Rhine was fully canalized between 1970 and 1977. The barrages Gamsheim and Iffezheim were installed to reduce river bed erosion processes; they are also used for hydropower generation. No further barrages are installed downstream of Iffezheim.

The international stakeholders developed a sophisticated feedback control strategy for this part of the river by taking into account multiple objectives and constraints. The intended modeling of this strategy in a hydraulic model by the German Federal Institute of Hydrology (BfG) turned out to be infeasible with the available modeling package SOBEK (Stelling & Duirmeijer 2003) due to several limitations. Technical limits restrict the maximum number of triggers and controllers per hydraulic structure. Conceptual limitations include the absence of deadbands in triggers and controllers, the consideration of external disturbances in PID controllers (Equation (2)), among others. The most severe drawback, however, turned out to be the SOBEK feature to assign control directly to hydraulic structures. This approach is not suitable in this case, because the Upper Rhine system is controlled in an integrated way, i.e., most control decisions affect multiple structures, for example, when the whole system is switched into flood mitigation mode. This means that it would be necessary to model the same decision process repeatedly for each affected structure. This is undesirable because it is error-prone and it makes the control model unnecessarily complex.

In a feasibility study, user programming for modeling the real-time control strategy was excluded as an option due to the need for dedicated code development, testing, and verification as well as the impossibility to split up the development of the modeling software and the model itself. This led to the decision to extend existing feedback control features in RTC-Tools and couple it to SOBEK on a time step base during runtime via OpenMI (see Gregersen et al. 2007). The coupling mechanism is described in detail by Becker et al. (2012b). Beside many extensions of existing feedback control features, the main conceptual advancement has been the introduction of binary decision trees to model the complex logic for the selection of the current operating mode of a structure and the related controller or operating rule. By definition, this method is fail-safe and leads to a unique selection. The final model of the Upper Rhine includes 51 decision trees (Schwanenberg et al. 2011) with about 150 triggers and 400 controllers or operating rules.

Figure 5 shows a close-up of the hydraulic model schematization. The node ‘Kehl–Kronenhof’ represents the corresponding gauge. Hydraulic model results of this node (Figure 6) are used to compute the crest level for the hydraulic structure ‘Kehl-Strasbourg’ according to the triggers and rules of the feedback control model. Within the Upper Rhine control strategy, there are three main operating modes relevant for the weir Kehl-Strasbourg, namely the standard mode (0), the special operating mode (1) and the flood retention mode (2). At the beginning of the simulation period, the weir operates in the standard mode. The target water level, i.e., the set point, of the weir is 140 m above sea level. It is conditional on the discharge provided by the hydraulic model and changes gradually to a target level of 139.7 m in preparation to an approaching flood after the up-crossing of threshold 1 at the gauge Kehl–Kronenhof in time step 2,330. Due to the exceedance of the second threshold at time step 2,660, the operating mode is switched to flood retention (mode 2) to store water in the barrage section upstream of the weir. Therefore, the set point is increased dynamically according to the discharge at the gauge Kehl–Kronenhof.

**MPC for the NZV polder system**

The regional water authority Noorderzijlvest (NZV) manages surface water resources in an area of about 1,440 km²...
in the north-east of the Netherlands. The system is a typical Dutch polder system, i.e., a low-lying region of former flood plains and marshes separated from the sea by dikes. The drainage of the region is mainly conducted artificially by pumping and, depending on the hydraulic conditions during low tide periods, supported by gravity flow through gates. The capacity of these structures and the related canal network is sufficient for a broad range of flood
events originating from heavy rainfall events in the region, but may become overloaded by extreme events such as occurred in the years 1998 and 2012 (see Becker et al. 2012a) causing potential flood inundation and dike failures. Since flood safety is under stress due to land subsidence on account of gas extraction and an expected sea level rise due to climate change, the NZV water authority is required to implement new flood protection measures in the near future.

Advanced real-time control techniques based on MPC have been identified as a potential non-structural measure for improving flood safety in an intelligent way. It aims at a better utilization of existing infrastructure by integrating rainfall and inflow predictions with an internal model of the water system and optimization schemes. This enables the in-time detection of rainfall events for pre-releasing water and creating additional storage. Another objective is the realization of energy and cost savings. These can be achieved by tactical deviation from the water level set points for exploiting gravity flow through gates and shifting pumping activities to night hours with less expensive energy tariffs. Although this objective is not related to flood events but to daily flow conditions, it is expected to increase the acceptance of the approach and demonstrate its capability to cover multiple objectives in an integrated way.

The representation of the regional water system of NZV in MPC requires a coarse internal model (Figure 7), in particular for bookkeeping of water volumes and related mean water levels in the different compartments of the system. Smaller compartments such as Electra 1 and Electra 2 or Lauwersmeer with minor water level gradients in the canal network can be modeled with single storage nodes. The larger compartments Electra 3 and Fivelingo require a

![Figure 6](image1.png)

**Figure 6** Simulation results of the conjunctive simulation of feedback control and hydraulics: discharge at gauge Kehl-Kronenhof (top), time-dependent operating mode (2nd from top), target water level upstream of weir Kehl-Strasbourg (2nd from bottom), upstream water level and crest level of weir Kehl-Strasbourg (bottom).

![Figure 7](image2.png)

**Figure 7** Overview of the internal model schematization for the water system of the NZV water authority operational area.
more detailed approach with three storage nodes each for taking into account the water level gradients between the central node of the compartment (representing the mean water level) and the upstream nodes at the outlet structures. This is important in particular, because of the limitations of the canal system upstream of the outlets at low water levels. These restrict the release to values below the maximum pump capacity. The nodes in the larger compartments need to be connected by hydraulic flow branches with the ability of modeling bidirectional flow and backwater effects. Given these requirements, we choose a diffusive wave model as an appropriate representation of the water system.

The pump and gate control flows are physically restricted by the pump capacities as well as the gate discharge capacities during low tides. Since the inflow into the water systems primarily depends on precipitation, the water levels depend on the severity of a flood event. It is therefore inappropriate to bound it by hard constraints, since this may lead to an infeasible solution of the optimization problem. An alternative is the definition of soft constraints, which penalize the up-crossing of a threshold without restricting the solution space. If constraints therefore apply only to control variables, the sequential MPC of Equation (6) turns out to be the most efficient set-up for this problem class. The objectives of flood control and combined energy and cost savings are translated into objective function terms of the following type:

\[
J = \sum_{k=1}^{N} \left( w_{up} (h^k - h_{sp})^2 + w_{up} \cdot \text{max}(h^k - h_{up}, 0)^2 + w_{down} \cdot \text{min}(h^k - h_{down}, 0)^2 + w_{h^k} \cdot Q^k + w_{\Delta Q^k} (\Delta Q^k)^2 + w_{\Delta d^k} (\Delta d^k)^2 \right)
\]

where the first term penalizes deviations of the water level \( h \) from set point \( h_{sp} \), the next two terms define a soft constraint for water level leaving an acceptable range \([h_{up}, h_{down}]\), the fourth term implements a time-dependent penalty \( w_{h^k} \) on pumping \( Q^k \) in relation to current energy costs (daily and nightly energy costs are 13 and 10-Cent/kWh, respectively). \( d \) denotes a gate position. Gate releases are not penalized and therefore preferred. The operational state of gates and pumps should not change too often, so the last two terms take care of smoothing the control trajectory by penalizing the increments \( \Delta Q^k = Q^k - Q^{k-1} \) and \( \Delta d^k = d^k - d^{k-1} \).

The control horizon of the MPC is 48 h with a time step of 5 resulting in 576 time steps. The MPC includes a total number of 17 aggregated structures, namely 13 pumping stations and four gate complexes, that leads to an optimization problem of 9,792 dimensions. The MPC has been applied in closed loop to the control of the internal model itself (‘RTC only’ option) and a more detailed, one-dimensional SOBEK model (‘SOBEK + RTC’ option) as a real-world replacement over an annual period from January 1 to December 31, 2007. The year 2007 has been chosen because of the consideration as a typical year in the hydrological sense, it contains phases of water scarcity and high water. Figure 8 presents the water levels achieved by feedback control (‘SOBEK feedback’ option) and the two MPC options both for flood events and normal flow conditions.

The flood control (Figure 8(a)) underlines the advantage of the MPC against feedback control. Whereas the feedback controller does not start to react to the flood event until the water level begins to rise, the MPC conducts a pre-release for creating additional storage and balancing the set point deviations to the negative and positive side. During normal flow conditions (Figure 8(b)), the MPC makes use of available storage for shifting pump activities into the off-peak hours at night. This results in a steadily increasing water level during the day and a decreasing one at night within the accepted range. The control performance with respect to the internal model itself and the more detailed SOBEK model is almost identical, which validates the internal model.

The energy and cost savings are shown in Table 1 for the compartments Electra 1 (only electrical pumps) and Five-lingo (electrical and diesel pumps, gates) representing the compartments with the minimum and maximum savings. It is obvious that even an advanced controller does not impact on the amount of pumped water, if pumps as in Electra 1 are the only release option. Therefore, the cost savings of 7% exclusively result from exploiting the energy price difference of 3-Cent/kWh between day and night. The Five-lingo compartment offers better conditions for savings, because of the available gates and the combination of electrical and diesel pumps. Table 1 shows a release shift towards the free-of-charge gates, and then followed by the
Figure 8 | Comparison of water levels in the Electra 1 polder achieved with feedback and predictive control for (a) a flood event and (b) under normal meteorological conditions.

<table>
<thead>
<tr>
<th></th>
<th>Volume [hm³]</th>
<th>Costs [Euro/hm³]</th>
<th>Total costs</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Electra 1, pumps</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feedback control (SOBEK)</td>
<td>55.1</td>
<td>47.5</td>
<td>102.6</td>
<td>21,485</td>
</tr>
<tr>
<td>MPC, RTC-Tools internal model</td>
<td>29.9</td>
<td>71.7</td>
<td>101.6</td>
<td>11,657</td>
</tr>
<tr>
<td>MPC, RTC-Tools and SOBEK</td>
<td>31.8</td>
<td>71.3</td>
<td>103.1</td>
<td>12,389</td>
</tr>
<tr>
<td>Fivelingo, pumps</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feedback control (SOBEK)</td>
<td>69.3</td>
<td>58.9</td>
<td>128.2</td>
<td>185,283</td>
</tr>
<tr>
<td>MPC, RTC-Tools internal model</td>
<td>43.5</td>
<td>45.0</td>
<td>88.4</td>
<td>113,433</td>
</tr>
<tr>
<td>MPC, RTC-Tools and SOBEK</td>
<td>45.5</td>
<td>48.7</td>
<td>93.2</td>
<td>116,963</td>
</tr>
<tr>
<td>Fivelingo, gates</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feedback control (SOBEK)</td>
<td>8.2</td>
<td>7.8</td>
<td>16.1</td>
<td>0</td>
</tr>
<tr>
<td>MPC, RTC-Tools internal model</td>
<td>28.7</td>
<td>25.1</td>
<td>53.8</td>
<td>0</td>
</tr>
<tr>
<td>MPC, RTC-Tools and SOBEK</td>
<td>28.1</td>
<td>24.2</td>
<td>52.3</td>
<td>0</td>
</tr>
<tr>
<td>Fivelingo, total</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feedback control (SOBEK)</td>
<td>77.5</td>
<td>66.7</td>
<td>144.2</td>
<td>185,283</td>
</tr>
<tr>
<td>MPC, RTC-Tools internal model</td>
<td>72.2</td>
<td>70.1</td>
<td>142.2</td>
<td>113,433</td>
</tr>
<tr>
<td>RTC/SOBEK</td>
<td>73.6</td>
<td>71.9</td>
<td>145.5</td>
<td>116,963</td>
</tr>
</tbody>
</table>
electrical pumps. Furthermore, the energy price difference contributes to the overall cost savings of 35%. Total annual cost savings in the complete system are in the order of 150,000 to 200,000 €.

MPC for scheduling the FCRPS

This case study focuses on the integrated short-term management of hydropower generation over a forecast horizon of 14 days for the Federal Columbia River Power System (FCRPS) in the Pacific Northwest of the USA. The river basin spans parts of seven western states of the USA and much of south-eastern British Columbia in Canada. It is an extensively developed system containing 31 multi-purpose federal hydropower projects and several others owned by non-federal entities including Canada. The FCRPS covers the federal projects, has a total installed capacity of about 22 GW and is operated jointly by the US Army Corps of Engineers, US Bureau of Reclamation, and Bonneville Power Administration (BPA). Figure 9(a) shows a general view of the major projects within the basin. The system is a cascading network of reservoirs subject to local and system-wide operating objectives and constraints. Typical operational constraints include flood control, fish and wildlife, navigation, irrigation, recreation, and power generation. A subset of hydropower projects (Figure 9(b)), referred to as the ‘Big-10’, is subject to an optimization-based short-term management by BPA’s Power & Operations Short-Term Planning group for guiding hydro operations and marketing activities. The remaining federal projects are either operated by rule-curves or located on river systems that do not support optimized releases on any particular operating day.

A main concern of BPA’s short-term operation of the Big-10 system is the ability to manage high-priority objectives, e.g., environmental obligations, flood control, reliability, and safety of the power network. Furthermore, the system optimization must enable BPA to assess potential over-generation supply conditions, assist in the refinement of mitigation strategies, and maximize the value of the FCRPS through net secondary revenue sales in the short-term planning horizon.

We present deterministic optimization results in a period with a focus on the chum salmon (Oncorhynchus keta) spawning operation of the FCRPS. The Columbia River population of the chum salmon has been listed as a threatened species under the US Endangered Species Act.

Figure 9 | (a) Map of the Columbia River Basin and selected hydro electric projects; (b) schematic overview of the Big-10 projects (federal) and non-federal projects in the middle Columbia River.
of 1973. They typically spawn in near-shore and tributary gravel bars in the lower Columbia River below Bonneville Dam. It is necessary to release enough water from Bonneville Dam at all times during the active spawning period (November through December) for adult chum to access favorable spawning habitat. Once the salmon establishes spawning nests (redds), the redd must remain submerged until they emerge in March and April. Therefore, it is desirable to release no more water than is absolutely necessary for the salmon to access such habitat in order to reduce the risk of chum establishing redd that are too high and likely to be dewatered prior to emergence. Nominally, this requires a discharge of 125 kcfs (ca. 3,540 m$^3$/s) to maintain the proper water surface elevation downstream of Bonneville Dam. Typically, the operational goal is to release the fixed outflow from Bonneville Dam while utilizing upstream storage to regulate and absorb fluctuations. Natural flows may be greater or less than the amount necessary to support chum spawning. The fixed outflow constraint is only modified within coordinated limits when it is demonstrated that upstream storage is insufficient to reduce flows to the fixed amount. Releases from Grand Coulee Dam and management of available storage in the lower Columbia River projects are used to regulate flows such that Bonneville Dam can release at the required rate in all hours. Spill is not desirable at any hydro project because of the economic loss and potential increase in total dissolved gases levels that are detrimental to the aquatic ecosystem.

For this scenario, the objective is to assess the potential to maximize weekday (Monday–Friday) heavy load hour (HLH) generation while meeting the defined constraints. Heavy load hours are defined as the 16-hour period starting at 06:00 and ending at 22:00. We focus on the feasible power surplus range within the constraints by defining an objective function with a quadratic penalty on deviations between load plus power surplus and hydropower production. The optimization problem is highly constrained in terms of control inputs, states, and model outputs. Therefore, the choice of a multi-shooting set-up is obvious. Besides the total project outflows (control), we select the forebay elevation (state) as additional optimization variables and implement the mass balance of the reservoirs as equality constraints. For a forecast horizon of 14 days with hourly time steps we obtain an optimization problem of 6,719 dimensions (forebay elevation plus total outflow times 10 reservoirs times 336 time steps minus 1 fixed forebay elevation). The pool routing equations result in 3,560 equality constraints with 28,636 non-zero elements in the equality constraint Jacobian. Environmental and power network constraints add another 25,872 inequality constraints with 72,583 non-zero elements in the inequality constraint Jacobian. The embedded IPOPT 3.11.0/HSL 2013 optimizer solves the optimization problem on a PC (Intel i5-3230M @ 2.60 GHz) in about 4 s, if the load balance is fulfilled and the objective function value becomes zero. If the load balance gets violated and more constraints become active, the execution time increases to about 30–60 s.

The optimization results indicate full compliance with the load balance for a power surplus range in HLH between 0 and 2,700 MW. A visual assessment of simulated data and constraints shows two main bottlenecks in the FCRPS, which limit the additional surplus beyond this point. One is the maximum drawdown limit at Grand Coulee reservoir of 1.5 ft in a moving time window of 24 hours (Figure 10(a)). Since higher hydropower production in the cascade of the downstream run-of-the-river power plants in the Columbia River depend on additional flow, the limitation of additional water extraction from Grand Coulee limits the overall amount of generated energy. Another bottleneck is the maximum turbine capacity of McNary, which is smaller than other projects in the cascade (Figure 10(b)). In combination with the requirement of avoiding spill, it limits the total flow through the cascade of projects in the lower Columbia River. Furthermore, it reduces the system capacity for daily peaking in case of high power surpluses. Note that peaking is only possible if the maximum turbine capacity of a project is higher than the required average flow. If both are the same, the turbine must run on full capacity continuously.

A power surplus of 2,800 MW and beyond leads to a violation of the load balance objective (Figure 10(c)). Power generation deficits occur mainly in the second week of the forecast horizon. Because of the quadratic penalty of load balance deviations, deficits get distributed over both the HLH and light load hours. The trade-off between them is configurable by weighting factor or can get limited by bound constraints. Figure 10(d) shows that the power production in the second week seems to hit a physical limit.
The realized power surplus for a surplus target of 2,800 and 2,900 MW is nearly the same in both options.

Note that surpluses assume the utilization of the complete FCRPS installed capacity.

**SUMMARY, DISCUSSION, AND OUTLOOK**

The novel RTC-Tools software package is a toolbox for modeling real-time control and decision support in water resources systems. It integrates various control techniques such as feedback control and MPC.

For modeling feedback control with RTC-Tools in combination with hydraulic models, we initiated a paradigm change away from model-specific control options inside of hydrologic and hydraulic models or the implementation of operating rules by user-programming towards a generic software package for the control component that runs coupled with a hydrologic or hydraulic model. The open modeling interface OpenMI supports this initiative, because RTC-Tools can be coupled basically with all hydraulic modeling packages that support this standard, no matter what commercial software, open source software, or research codes. Another important innovation of the presented framework for the practical modeling of complex feedback strategies found in the River Rhine case above as well as many highly regulated water systems in the Netherlands and elsewhere is the use of binary decision trees to select the appropriate feedback controller in a fail-safe way.

Whereas feedback control has already reached a high maturity level, MPC is subject to ongoing research.

![Figure 10](image-url)
RTC-Tools, making MPC available as a generic tool in water resources applications, helps to bring MPC to practitioners and to establish it as common practice. Possible benefits of model MPC against the feedback control method have been illustrated in the application cases described in the previous two sections, and it has been shown that the new framework can be efficiently applied to large-scale water systems in a real-time decision support context.

Note that the advantage of MPC as an online optimization approach significantly depends on the skill of hydrological forecasts. If we anticipate on wrong forecasts, the resulting control may cause harm (Bemporad & Morari 1999). Therefore, an important field of future research is the explicit consideration of forecast uncertainty by probabilistic ensemble forecasts (Raso et al. 2013) and the extension of the deterministic optimization towards a multi-stage stochastic optimization to improve the control robustness and decision support. The enhancement of the computational performance of the package is also a focus to enable the large-scale application of RTC-Tools, for example, on a national scale in the Netherlands to manage water extractions (Talsma et al. 2013).

REFERENCES


First received 24 March 2014; accepted in revised form 4 July 2014. Available online 9 September 2014