



Wehrsteuerung in HydroAS

Entwicklung und Anwendung eines Wehrsteuerungsmoduls zur Abbildung der Kraftwerkssteuerung an der österreichischen Donau

Achim Naderer und Johannes Rohde, HydroAS Anwendertreffen 2022

Agenda

- ▶ Scripting mit HydroAS
- ▶ Benefits für den HydroAS-Kern aus diesem Projekt
- ▶ Technische Umsetzung mittels LUA
- ▶ Modellierung des Wehrbereiches
- ▶ Anwendungsbeispiel / Ergebnisse



- ▶ Abfragen & Verändern von Daten während der Simulation
 - ▶ Bauwerkssteuerung
 - ▶ Zufluss-Zeitreihen
 - ▶ Interaktion mit anderen Modellen
 - ▶ Deich-/ Dammbrüche
 - ▶ Spezielle Auswertungen
 - ▶ Variantenuntersuchungen

- ▶ Skriptsprache LUA

- ▶ weiteres Infos:
 - ▶ Kap. 8 „Steuerung mit Scripting“ im Benutzerhandbuch
 - ▶ Vortrag Anwendertreffen 2020 „Einsatz von Scripting und die Modellierung von Steuerungen“



Benefits für den HydroAS-Kern aus diesem Projekt (seit 5.2)

- ▶ Erweiterung der Scripting-Schnittstelle für alle HydroAS Anwender*innen:
 - ▶ Scripting Zugriff auf nahezu alle Eingabedaten zu allen Auslaufrandbedingungen
 - ▶ Scripting Zugriff auf interne Daten von HydroAS: Relaxation-Faktor/ Proportionalitäts-Faktor für die Übertragung einer Oberwasserpegelabweichung auf das Wehr

 - ▶ Mit Scripting abfragbar sind alle Daten, die in SMS eingegeben werden sowie die Daten zum Nodestring (Wieviele/Welche Knoten?)
 - ▶ Mit Scripting änderbar sind die meisten Daten, die in SMS über das Randbedingungsfenster eingegeben werden können
 - ▶ Insbesondere Scriptable: Zeitreihen und andere Funktionen (W-Q-Beziehung,...)
 - ▶ Aktuelle Q am Abfluss-Rand abfragen
 - ▶ Auslauf H-Wehr: Maximale hydraulische Leistungsfähigkeit am Wehr abfragen

- ▶ Das ermöglichte im konkreten z.B.:
 - ▶ Abfluss-Q-Zeitreihen am Anfang der Simulation von Datei lesen
 - ▶ Auslaufrandbedingung (z.B. Auslauf Typ Energieliniengefälle) in Abhängigkeit von Modellzuständen verändern
 - ▶ Abfluss-Q in Abhängigkeit von Modellzuständen verändern

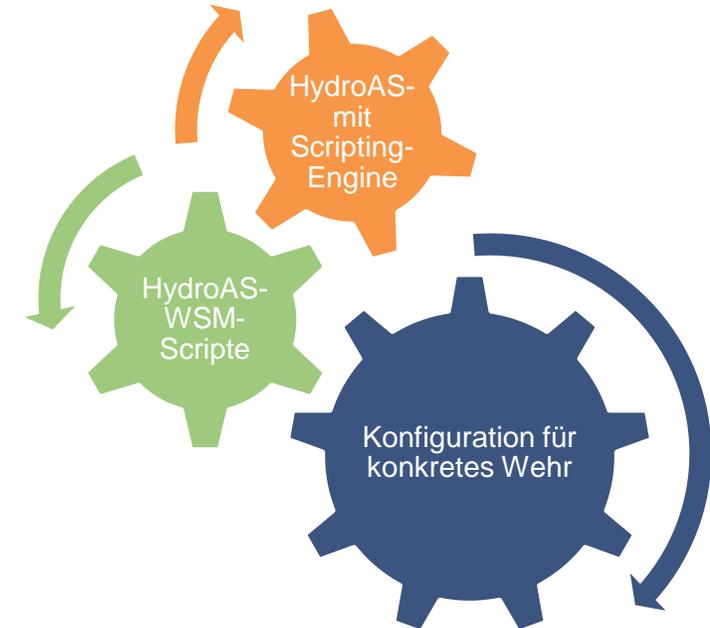


▸ Herausforderung:

- Erfassung aller verschiedenen Wehrbetriebsordnungen a. d. österr. Donau
- z.T. sehr komplex
- Einfache Nutzung durch die Endanwender*innen

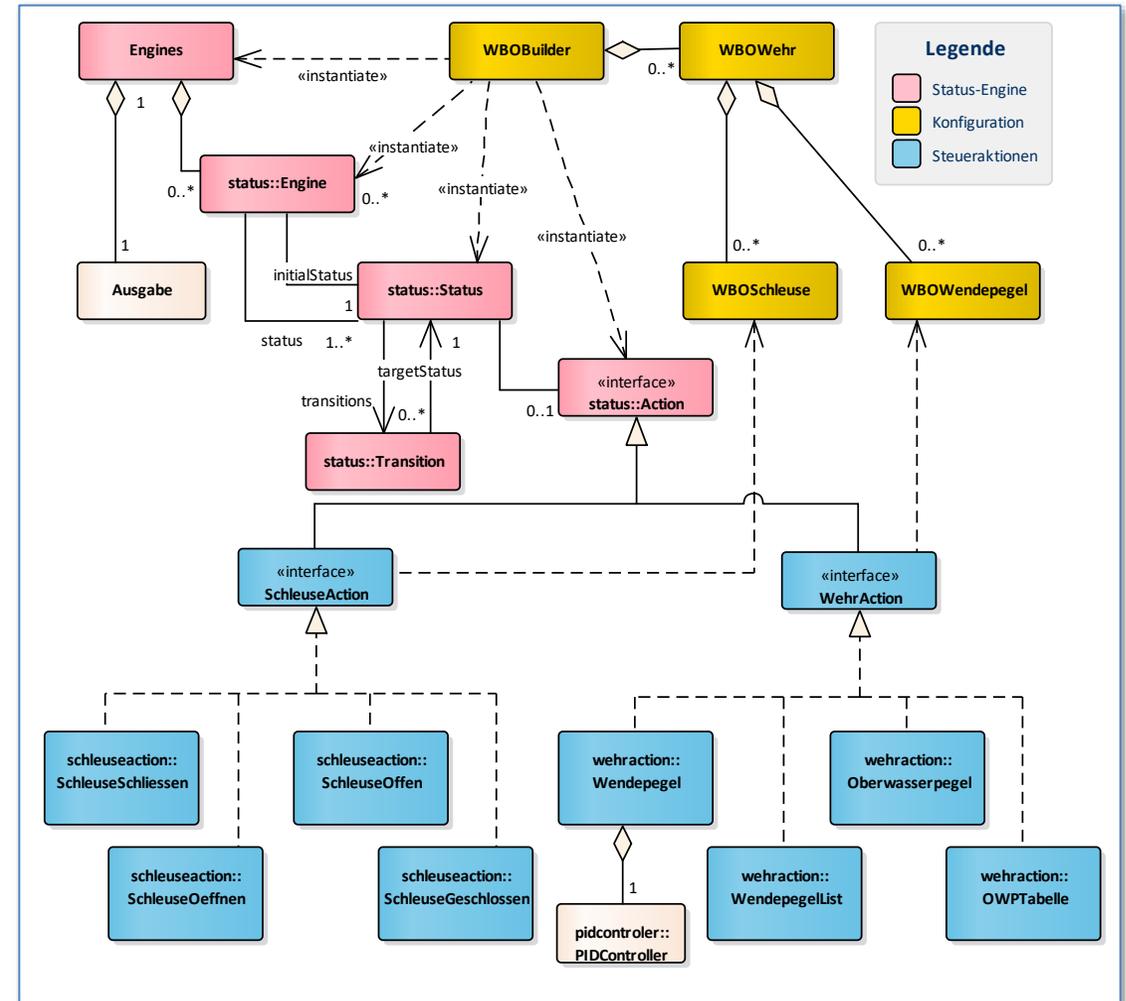
▸ Umsetzung

- Auslagerung der Skriptsteuerung in verschiedene LUA-Klassen „WSM-Skripte“
- Ausschließlich ein einzelnes Konfigurationsskript für die Endanwendung „Config.lua“



Technische Umsetzung mittels LUA

- Umfangreiches Steuerskriptes im Hintergrund
 - Aufbau in verschiedenen Klassen die
 - Steuervorgaben aus der WBO vorhalten
 - Wehrkonfigurationen abfragen
 - Daten aus HydroAS abfragen
 - Änderungen in den Randbedingung in HydroAS vornehmen
 - Ergebnis- und log-Dateien ausschreiben
- sehr komplex und allgemeingültig für die existierenden WBO an der österreichischen Donau



- Übersichtliche Config.lua für die Endanwendung
 - Proportionalitätsfaktoren
 - Angabe der Wehr-Parameter
 - Status aus der Wehrbetriebsordnung
 - Schleuseneinsatz im HW-Fall

- Während der Simulation wird
 - die Wasserspiegelganglinie des H-Wehrs überschrieben
 - die Höhe an den Schleusentoren geändert

```
WBOConfiguration.lua x
1  local wbo=require("wbo")
2
3  return {
4    wehr = {
5      name = "WehrH",
6      status = {
7        wbo.Oberwasserpegel(155),
8        wbo.Wendepegel("WendeP",158),
9        wbo.Oberwasserpegel(153),
10       wbo.Wendepegel("WendeP",160),
11       wbo.Oberwasserpegel(148)
12     },
13     schleuse = {
14       einlauf = "Schleusentor"
15     }
16   }
17 }
```

► Übersichtliche Config.lua für die Endanwendung

► Beispiel:

- Wehr am Nodestring „WehrH“
- Absenkgeschwindigkeit des Oberwasserpegels: max 30cm/h

► Status-Wechsel:

1. Oberwasserpegel halten auf 155 m.ü.A.
2. bis WSP am Wendepunkt 158 m.ü.A. überschreitet
3. Dann den Oberwasserpegel absenken und auf 153 m.ü.A. halten
4. bis Wendepunkt 160 m.ü.A. überschreitet,
5. Anschließend Oberwasserpegel absenken und auf 148 m.ü.A halten

► Schleuse:

- Wenn 85% der maximale Leistungsfähigkeit des Wehr erreicht überschritten wurden:
- Schleusen öffnen (Absenken der Schleusentore)
- Schleusen schließen wenn bei 50% Auslastung des Wehres
- Tor-Geschwindigkeiten, minimale Öffnungszeiten und Absenktiefe beachten

```
1 local wbo=require("wbo")
2
3
4 return {
5     kontrollzeitschritt = 60,           -- Länge des Kontrollzeitschrittes
6     wp_pid_faktI = 0.,                -- Integral Faktor des PIDControllers
7     wp_pid_faktD = 0.,                -- Differential Faktor des PIDControllers
8     wp_pid_faktP = 0.4,               -- Proportional Faktor des PIDControllers
9     wehr = {                           -- Anfang Konfiguration Wehr
10        name = "WehrH",                 -- Nodestringname des Wehrkörpers
11        max_absenkgeschwindigkeit_owp = (30/100)/3600, -- 30cm/h
12        toleranz_pos = 0.2,
13        toleranz_neg = 0.1,
14        status = {                       -- Mehrere WBO-Status
15            wbo.Oberwasserpegel(155),
16            wbo.Wendepunkt("WendeP",158),
17            wbo.Oberwasserpegel(153),
18            wbo.Wendepunkt("WendeP",160),
19            wbo.Oberwasserpegel(148)
20        },
21        schleuse = {
22            einlauf = "Schleuse",         -- NodeStrings, der/die beim Öffnen der Schleuse
23                                           -- abgesenkt werden
24            q_ausgabe = "SchleuseQ",     -- NodeStrings, die das Q messen
25            oeffnen_bei_proz = 85,        -- Schleuse öffnen bei diesem Prozentsatz des
26                                           -- max. möglichen Abflusses
27            schliessen_bei_proz = 50,     -- Schleuse schließen bei diesem Prozentsatz
28                                           -- des max. möglichen Abflusses
29            geschwindigkeit = 0.0005,    -- Absenkgeschwindigkeit
30            offen_bei = 132.15,          -- min Sohl-Höhe, auf die Einlauf gesenkt wird
31            minStatusZeit = 3600*12,     -- Nach dem Öffnen oder Schliessen der Schleuse
32                                           -- bleibt sie wenigstens die Zeit offen/geschlossen
33        }
34    }
}
```

► Übersichtliche Config.lua für die Endanwendung

► Abbildung weitere Fälle der WBO

► Status-Wechsel bei Tabelle WP, OWP:

1. OWP halten bis WP überschritten
2. Dann WP halten bis erster Wert aus der Tabelle erreicht
3. Tabelle steuert OWP und WP
lineare Interpolation zwischen den Werten

```
13 status = { -- Mehrere WBO-Status
14     wbo.Oberwasserpegel(155), -- Status 1: OWP halten
15     wbo.Wendepegel("WendeP",158), -- Status 2: Wendepegel halten
16     wbo.WP2OWPTable("SteuerP",{ -- Status 3: Tabellensteuerung
17         {160.9,152.2},
18         {161.0,151.0},
19         {161.1,150.8},
20         {161.5,150.0},
21         {162.1,146.0}
22     })
23 },
```

► Status-Wechsel beim gleichzeitigen Halten zweier WP:

1. OWP halten bis WP überschritten
2. Dann WP halten bis erster Wert

```
15 status = {
16     wbo.Oberwasserpegel(150.30), -- Stauziel des OWP
17     wbo.WendepegelList({"WP1",150.40},{ "WP2",150.60}), -- WP1 und WP2 gleichzeitig halten
18     wbo.Oberwasserpegel(149.20) -- OWP halten auf 149.20
19 },
```

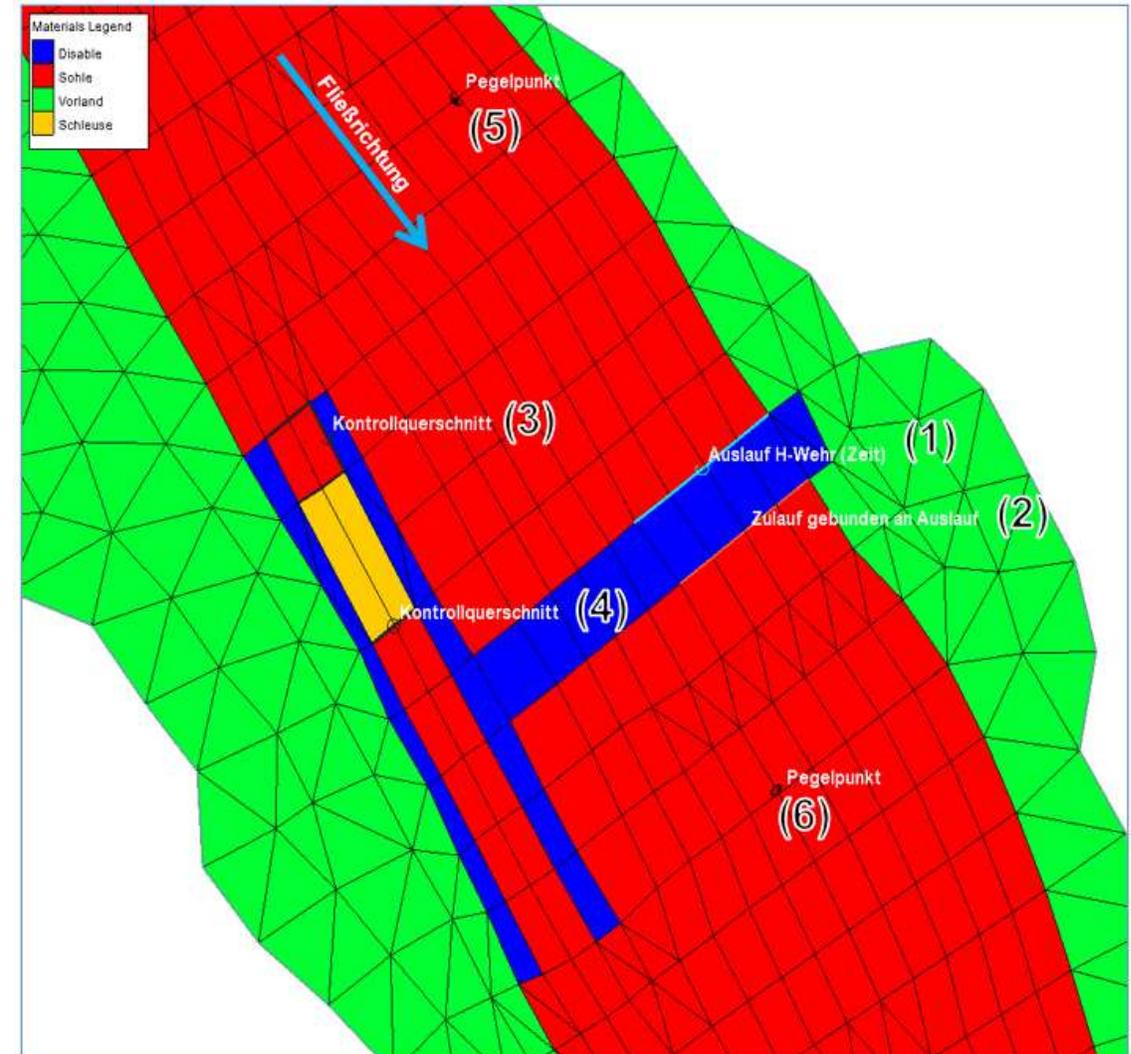
Modellierung des Wehrbereiches (vereinfachtes Netz)



► Abbildung des Wehrbereichs in SMS, Darstellung der notwendigen Randbedingungen

► Wendepegel weiter im Oberwasser (nicht dargestellt)

1. Definition des Wehrquerschnittes (Nodestringlänge muss Wehrbreite entsprechen) als H-Wehr Zeit
2. Zulauf gebunden an Wehr-Auslauf
3. Kontrollquerschnitt zur Definition der Schleusentore
4. Kontrollquerschnitt zur Bestimmung Q-Schleuse
5. Oberwasserpegelpunkt
6. Unterwasserpegelpunkt (Rückstau beeinflusstes Wehr)



Ergebnis-log-Datei (fiktives Beispiel)

► Informationen über alle Status, Wasserspiegel und Abflüsse über die Simulation

TS-Time [sec]	Wehr-AW_Wehr-Status	Schleuse-AW_Wehr (SchleuseAW)-Status	AW_Wehr-OWP-Ist [müA]	AW_Wehr-OWP-Soll [müA]	AW_Wehr-Q [m³/s]	AW_Wehr-QMax
0.279	Oberwasserpegel AW_Wehr 153.50	SchleuseGeschlossen AW_Wehr (SchleuseAW)	153.500	153.500	0.000	8907.234
30.194	Oberwasserpegel AW_Wehr 153.50	SchleuseGeschlossen AW_Wehr (SchleuseAW)	153.241	153.500	2.735	13672.011
315900.207	Oberwasserpegel AW_Wehr 153.50	SchleuseGeschlossen AW_Wehr (SchleuseAW)	153.510	153.500	4683.845	13588.104
315930.207	Oberwasserpegel AW_Wehr 153.50	SchleuseGeschlossen AW_Wehr (SchleuseAW)	153.510	153.500	4684.951	13587.998
315960.175	Wendepiegel WPTh 154.30	SchleuseGeschlossen AW_Wehr (SchleuseAW)	153.510	153.500	4686.060	13587.973
315990.175	Wendepiegel WPTh 154.30	SchleuseGeschlossen AW_Wehr (SchleuseAW)	153.510	153.499	4687.229	13587.841
482340.016	Wendepiegel WPTh 154.30	SchleuseGeschlossen AW_Wehr (SchleuseAW)	152.708	152.700	6256.605	12491.391
482370.047	Wendepiegel WPTh 154.30	SchleuseGeschlossen AW_Wehr (SchleuseAW)	152.708	152.700	6257.856	12491.378
482400.078	Oberwasserpegel AW_Wehr 152.70	SchleuseGeschlossen AW_Wehr (SchleuseAW)	152.708	152.700	6259.107	12491.302
482430.110	Oberwasserpegel AW_Wehr 152.70	SchleuseGeschlossen AW_Wehr (SchleuseAW)	152.708	152.700	6260.356	12491.208
609540.093	Wendepiegel PH 154.35	SchleuseGeschlossen AW_Wehr (SchleuseAW)	151.301	151.300	8994.607	10582.509
609570.155	Wendepiegel PH 154.35	SchleuseGeschlossen AW_Wehr (SchleuseAW)	151.301	151.300	8994.896	10582.507
609600.218	Wendepiegel PH 154.35	SchleuseOeffnen AW_Wehr	151.301	151.300	8995.174	10582.498
609630.093	Wendepiegel PH 154.35	SchleuseOeffnen AW_Wehr	151.301	151.300	8995.456	10582.476
613470.093	Wendepiegel PH 154.35	SchleuseOeffnen AW_Wehr	151.300	151.300	9030.311	10579.643
613500.155	Wendepiegel PH 154.35	SchleuseOeffnen AW_Wehr	151.300	151.300	9030.311	10579.632
613530.155	Oberwasserpegel AW_Wehr 151.30	SchleuseOeffnen AW_Wehr	151.300	151.300	9030.311	10579.655
613560.218	Oberwasserpegel AW_Wehr 151.30	SchleuseOeffnen AW_Wehr	151.300	151.300	9030.311	10579.659
690780.178	Oberwasserpegel AW_Wehr 151.30	SchleuseOeffnen AW_Wehr	151.295	151.300	5660.056	10812.163
690810.240	Oberwasserpegel AW_Wehr 151.30	SchleuseOeffnen AW_Wehr	151.295	151.300	5659.690	10812.200
690840.053	Oberwasserpegel AW_Wehr 151.30	SchleuseOffen AW_Wehr (SchleuseAW)	151.295	151.300	5659.326	10812.256
690870.115	Oberwasserpegel AW_Wehr 151.30	SchleuseOffen AW_Wehr (SchleuseAW)	151.296	151.300	5658.961	10812.351
757380.245	Oberwasserpegel AW_Wehr 151.30	SchleuseOffen AW_Wehr (SchleuseAW)	151.295	151.300	5405.203	10809.956
757410.120	Oberwasserpegel AW_Wehr 151.30	SchleuseSchliessen AW_Wehr (SchleuseAW)	151.295	151.300	5404.820	10809.957
757440.058	Oberwasserpegel AW_Wehr 151.30	SchleuseSchliessen AW_Wehr (SchleuseAW)	151.295	151.300	5404.451	10809.958

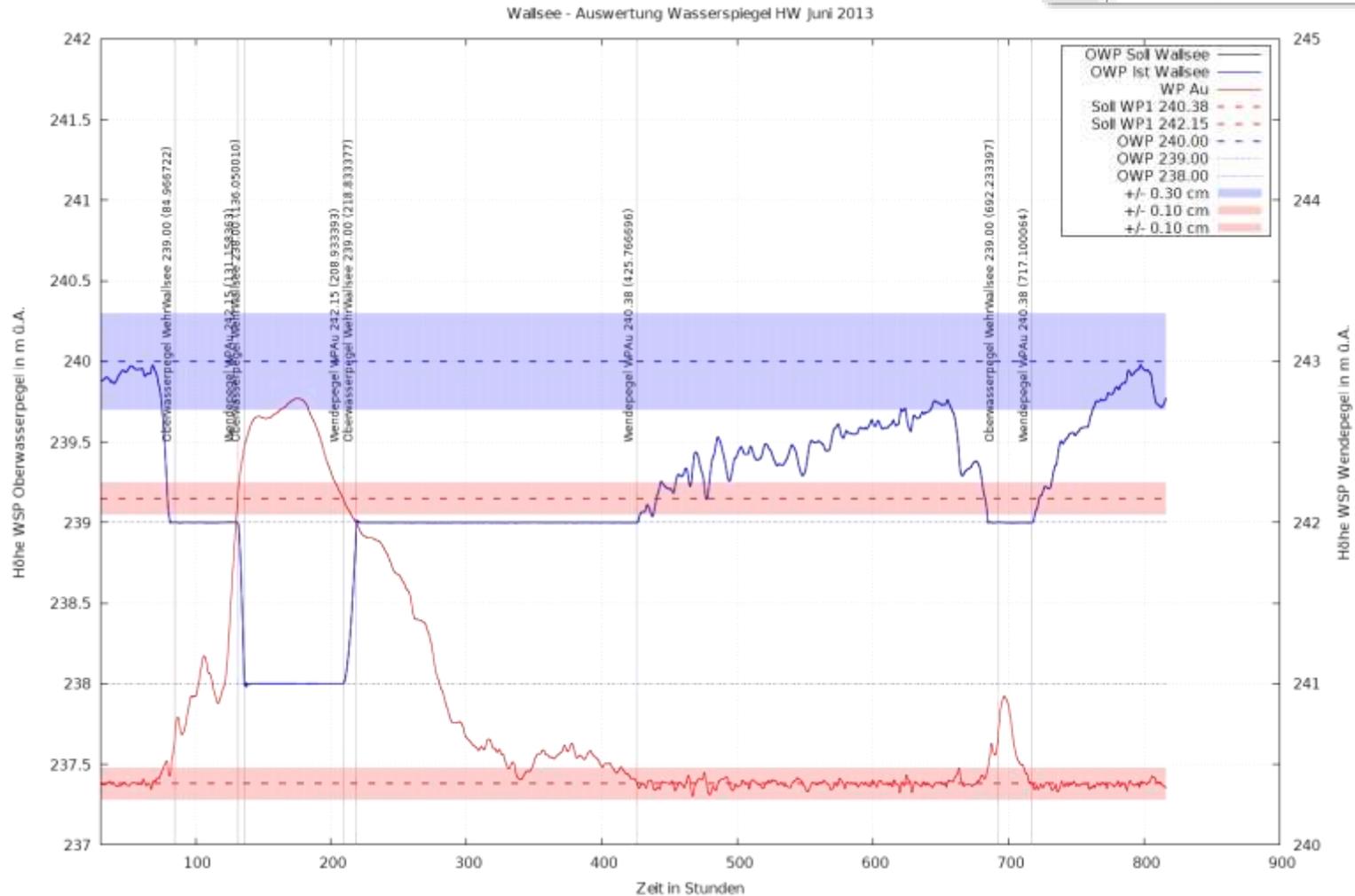
Validierung des Wehrsteuerungsmoduls

- ▶ WSM wurde auf Basis bestehender 2D-Modelle und beobachteter Abflussganglinien validiert.
- ▶ 2D-Modelle wurden für die Verwendung des WSM im Kraftwerksbereich überarbeitet.
- ▶ Es erfolgte keine Überarbeitung der grundsätzlichen Rauigkeitswerte (Kalibrierung).
- ▶ Fokus wurde auf die Abbildung der Wehrbetriebsordnung gelegt.
- ▶ Es handelt sich bei den Simulationen daher nicht um eine dedizierte Nachrechnung eines Hochwasserereignisses, da nur aktuelle Wehrbetriebsordnungen verwendet wurden.
- ▶ Validierung der Steuerungsskripte von 7 Kraftwerken mit jeweils 3 Abflussereignissen

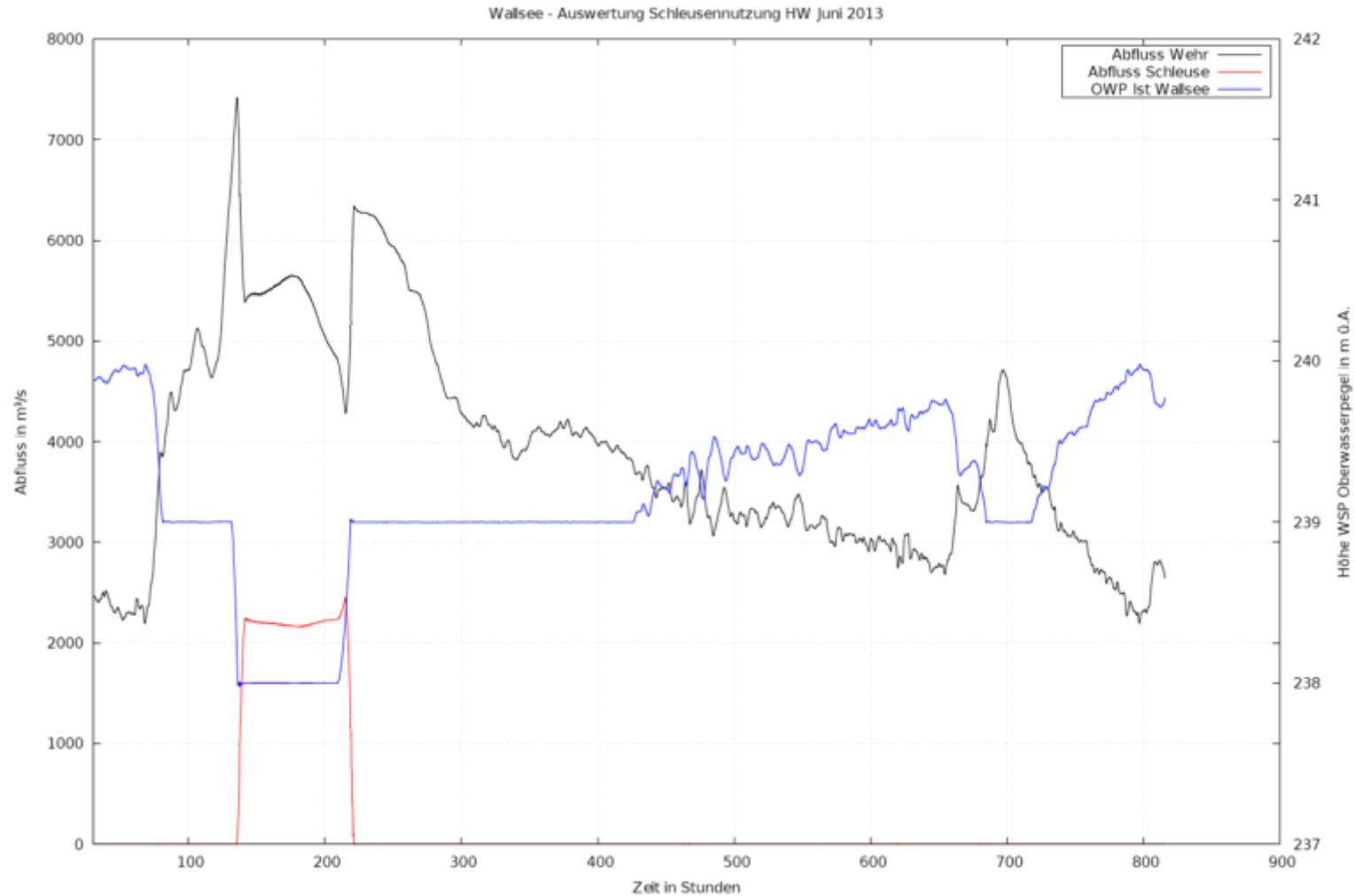
Anwendungsbeispiel Wasserspiegelganglinie

```

46 | status = {
47 |   wbo.Oberwasserpegel (240.0),      -- Stauziel des OWP
48 |   wbo.Wendepiegel ("WPAu", 240.38), -- WP Au auf 240.38 halten
49 |   wbo.Oberwasserpegel (239.00),    -- OWP auf 239 halten
50 |   wbo.Wendepiegel ("WPAu", 242.15), -- WP Au auf 242.15 halten
51 |   wbo.Oberwasserpegel (238.0)     -- OWP auf 238 halten
52 | },
    
```



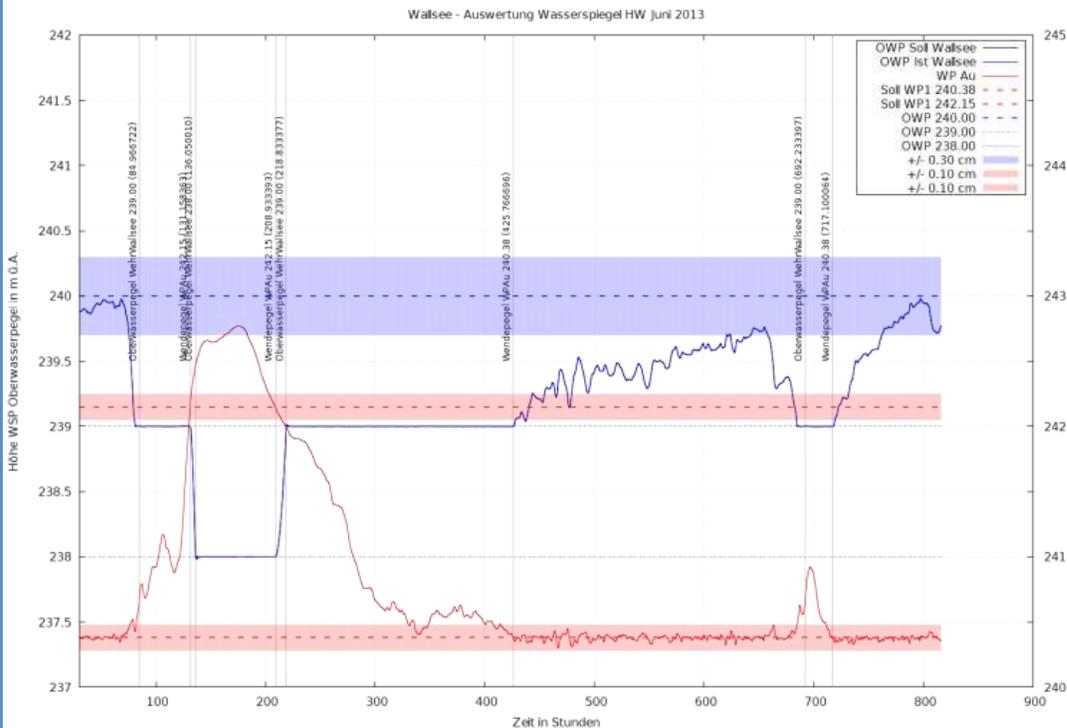
Anwendungsbeispiel Schleusennutzung



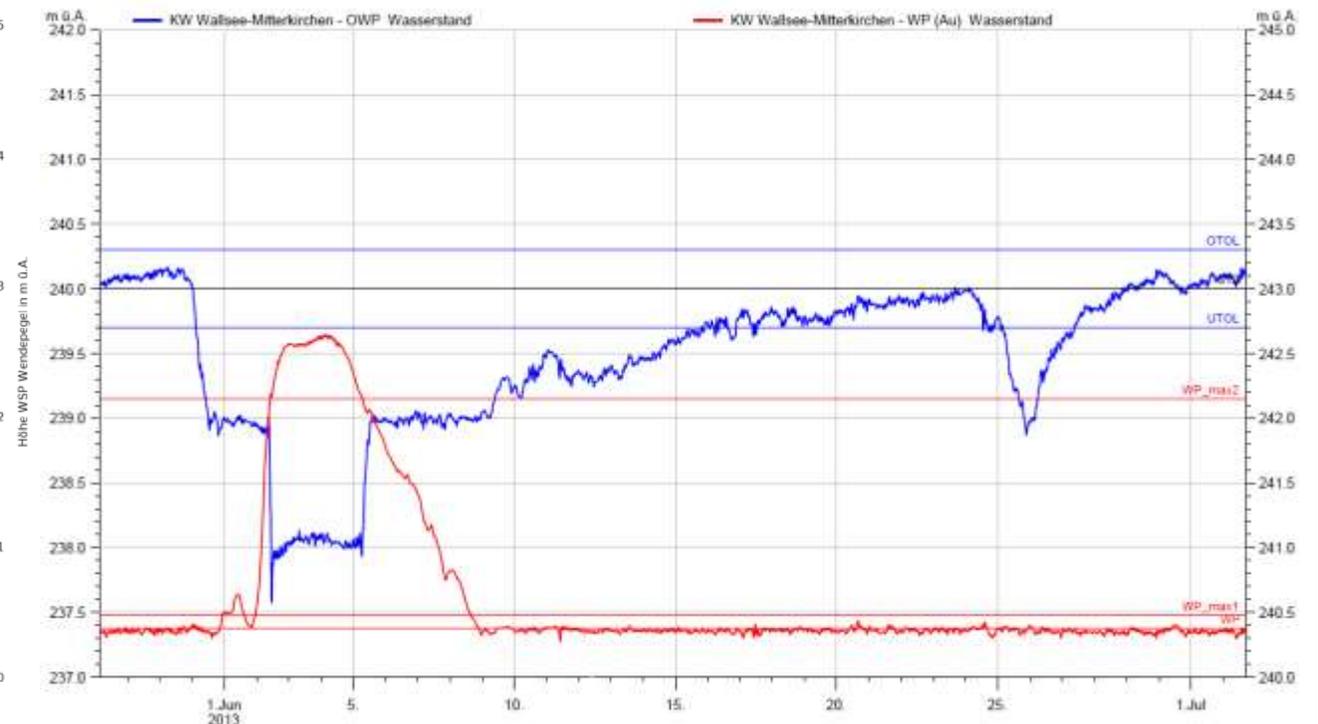
Anwendungsbeispiel Vergleich mit tatsächlichem Ereignis

- ▶ KW Wallsee-Mitterkirchen
- ▶ Simulation der Abflussganglinie des HW2013
- ▶ Ereignis 06/2013 Q_{\max} Pegel Grein 11000 m³/s

Simulation



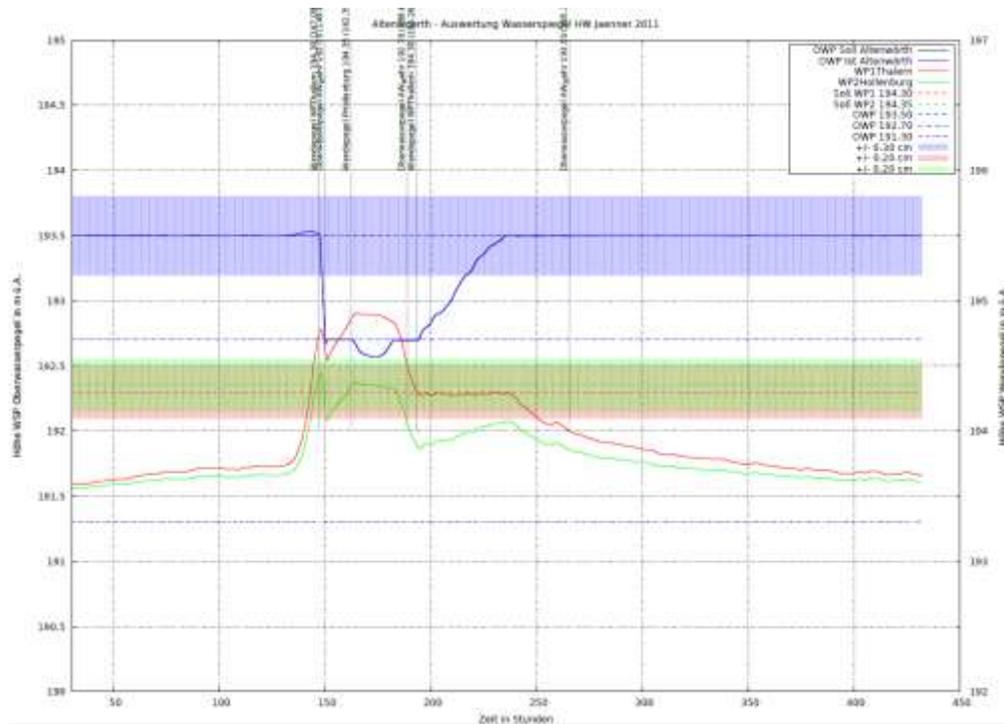
Beobachtung



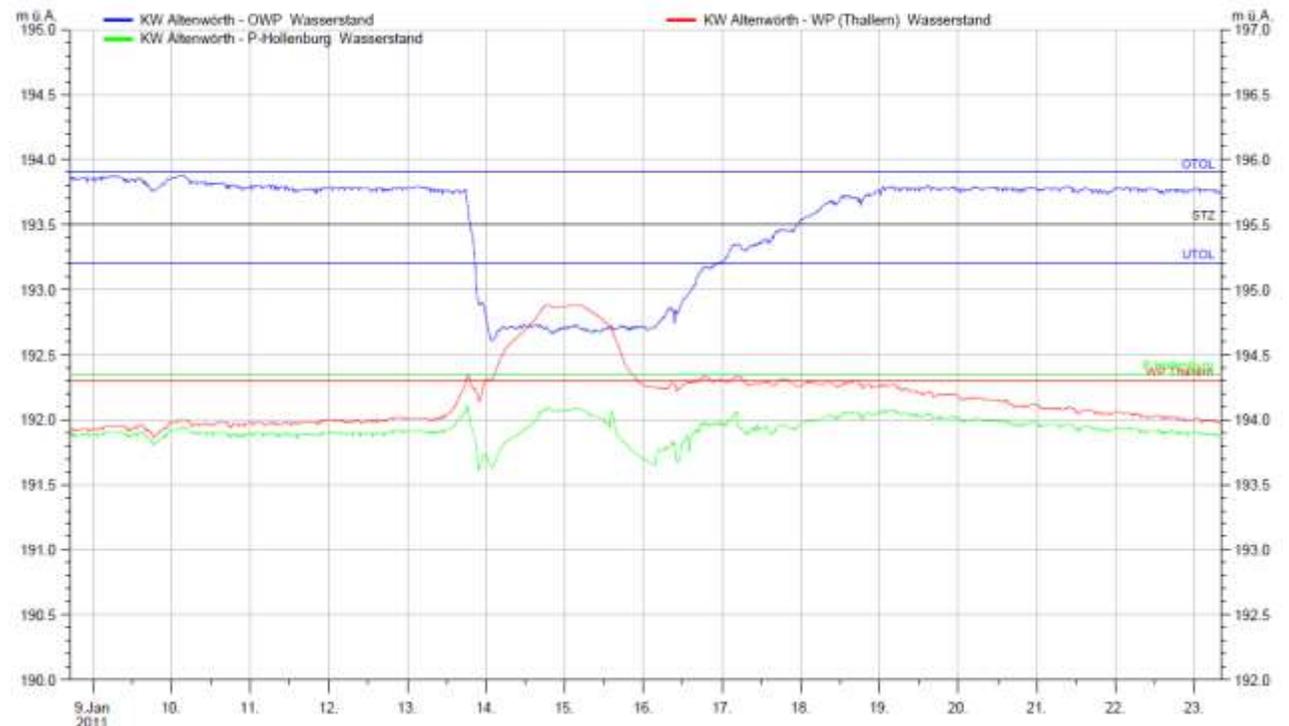
Anwendungsbeispiel Vergleich mit tatsächlichem Ereignis

- ▶ KW Altenwörth
- ▶ Simulation der Abflussganglinie des Jänner 2011
- ▶ Ereignis 01/2011 Q_{\max} Pegel Kienstock 7372 m³/s

Simulation

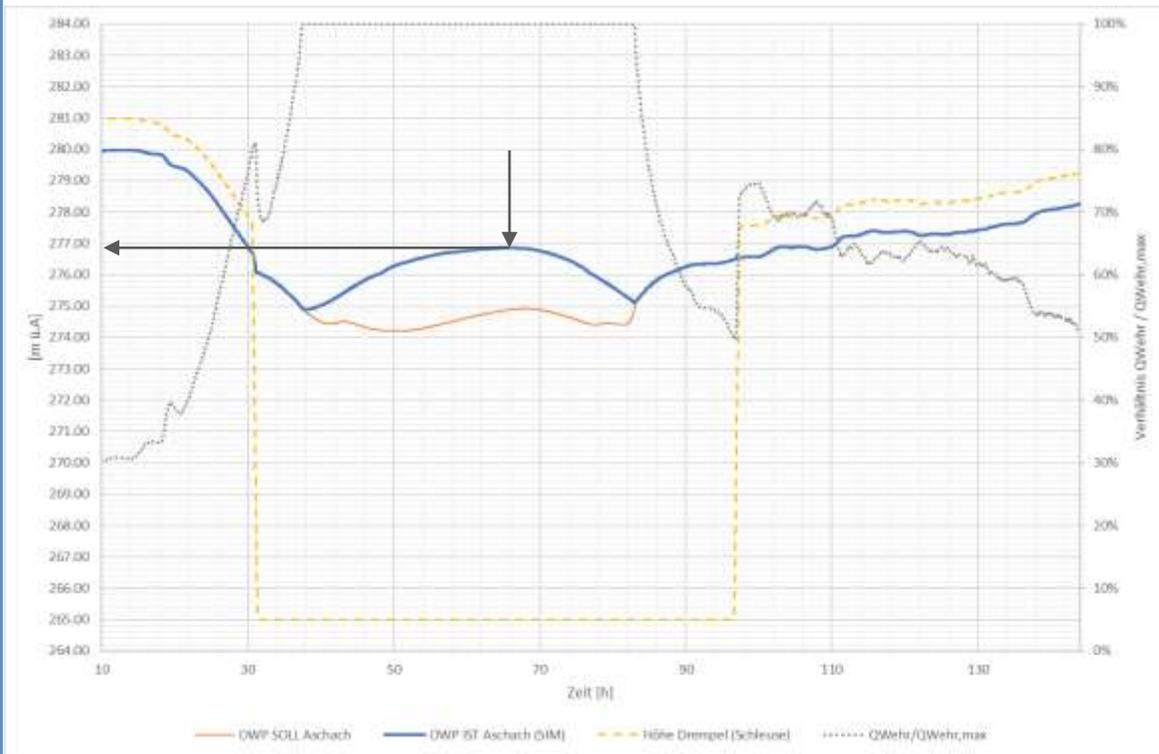


Beobachtung



Anwendungsbeispiel Vergleich mit tatsächlichem Ereignis

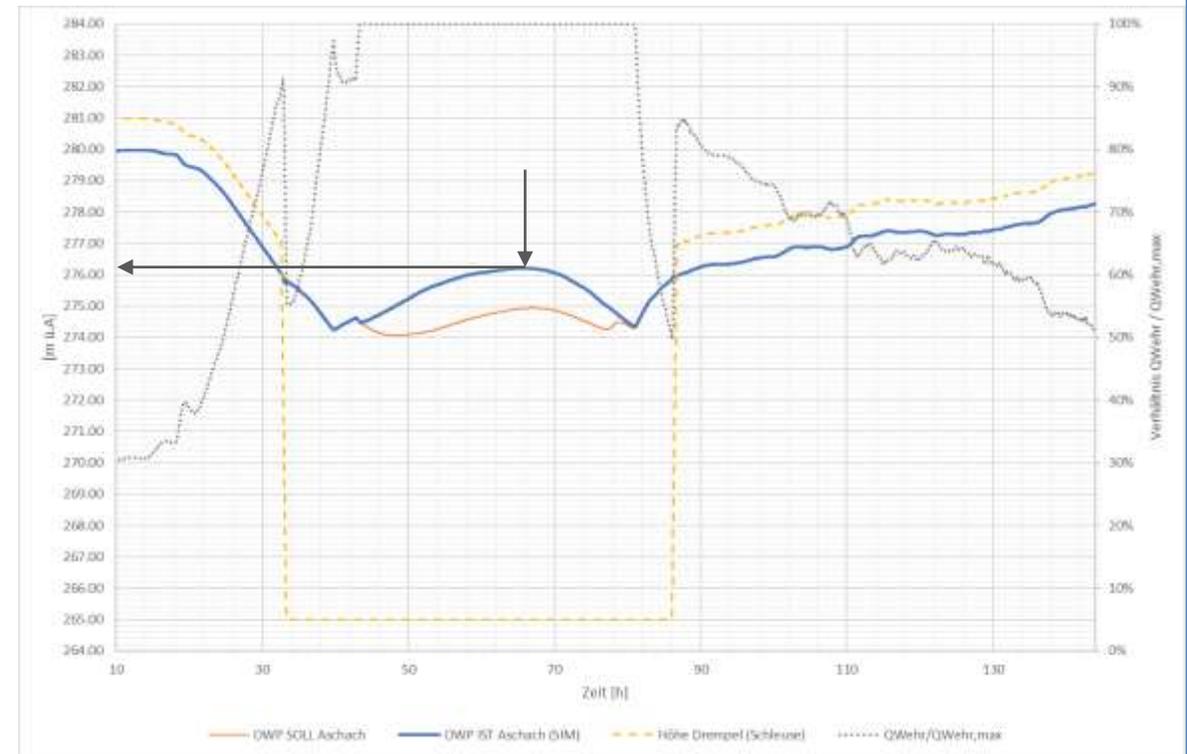
- ▶ KW Aschach
- ▶ Schleuse öffnen bei $Q_{\text{Wehr}} / Q_{\text{Wehr,max}} = 80\%$
- ▶ Keine Remobilisierung (Abtrag) im Bereich Schleusenzulauf berücksichtigt



Öffnen der Schleuse

Schließen der Schleuse

- ▶ KW Aschach
- ▶ Schleuse öffnen bei $Q_{\text{Wehr}} / Q_{\text{Wehr,max}} = 90\%$
- ▶ Remobilisierung (Abtrag) im Bereich Schleusenzulauf berücksichtigt



Öffnen der Schleuse

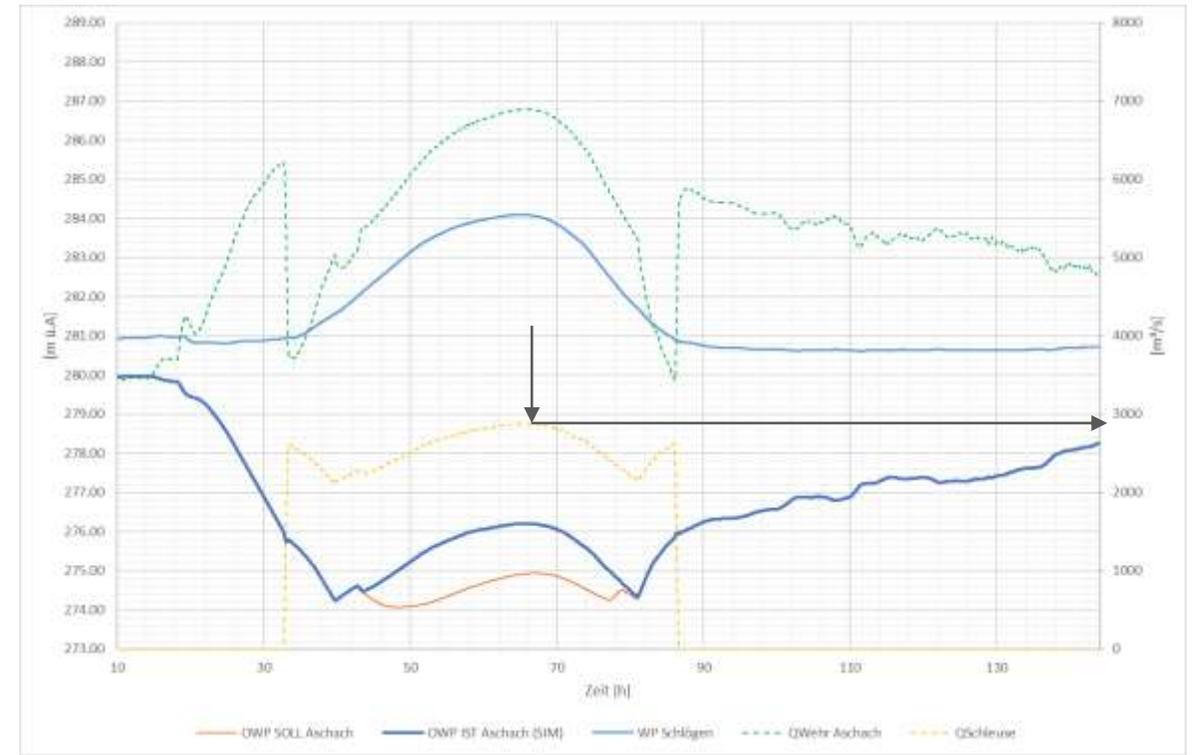
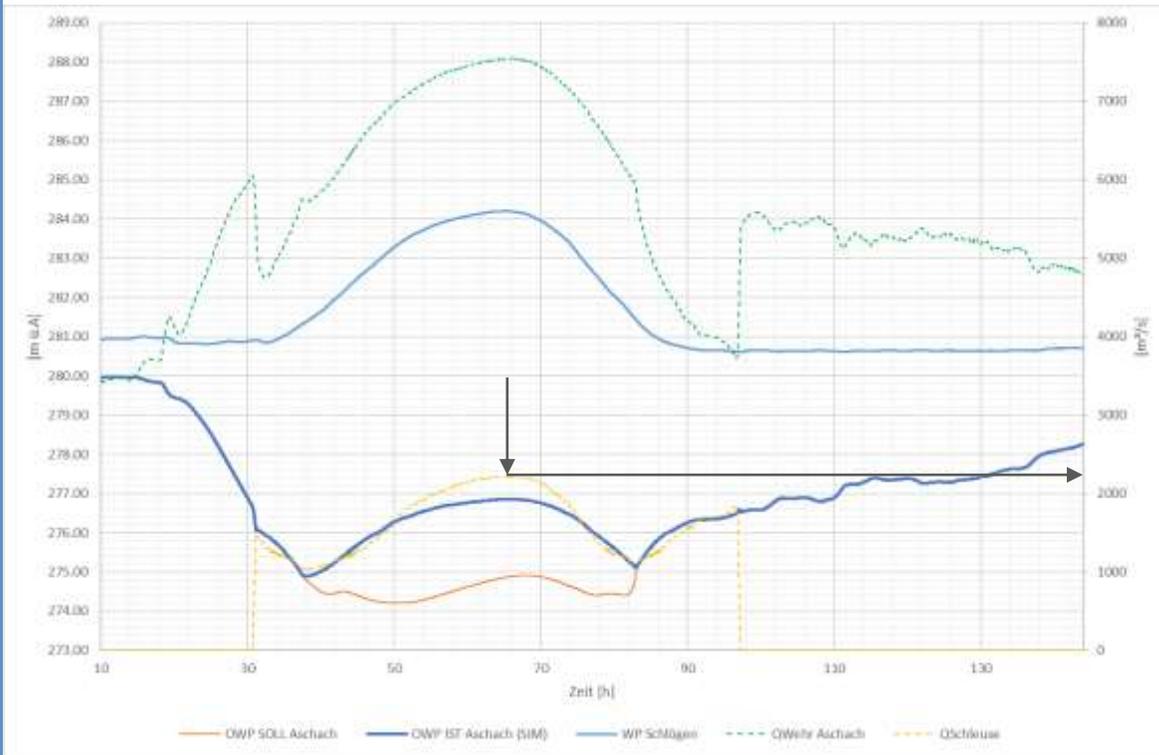
Schließen der Schleuse

14. September 2022

Anwendungsbeispiel Vergleich mit tatsächlichem Ereignis

- ▶ KW Aschach
- ▶ Schleuse öffnen bei $Q_{\text{Wehr}} / Q_{\text{Wehr,max}} = 80\%$
- ▶ Keine Remobilisierung (Abtrag) im Bereich Schleusenzulauf berücksichtigt

- ▶ KW Aschach
- ▶ Schleuse öffnen bei $Q_{\text{Wehr}} / Q_{\text{Wehr,max}} = 90\%$
- ▶ Remobilisierung (Abtrag) im Bereich Schleusenzulauf berücksichtigt



Öffnen der Schleuse

Schließen der Schleuse

Öffnen der Schleuse

Schließen der Schleuse

14. September 2022

18

Zusammenfassung

- ▶ Anforderungen an Wehrsteuerungsmodul aus wasserwirtschaftlicher Praxis
- ▶ Ziel der **realistischen Abbildung komplexer Wehrbetriebsordnungen** der Kraftwerke an der österreichischen Donau **in hydrodynamischen 2D-Modellen**
- ▶ Umsetzung auf Basis der freien Scripting-API in HydroAS „Lua“
- ▶ Erweiterung der Scripting-Schnittstelle für alle HydroAS Anwender*innen (ab HYDRO_AS-2D 5.2)
- ▶ WSM wurde auf Basis bestehender 2D-Modelle und beobachteter Abflussganglinien validiert (7 Kraftwerke mit jeweils 3 Abflussereignissen)
- ▶ Möglichkeiten der Wehrsteuerung mittels Scripting-API vielfältig und flexibel

Anmerkungen

- ▶ Hydrodynamische Simulationen erfolgten mit fester Sohle, d.h. keine Berücksichtigung von Remobilisierungen im Zulauf der Kraftwerke (Einfluss auf Abfluss Wehrfelder und Schleusen)
- ▶ Grenzen der 2D-Modellierung bei hoch turbulenten hydrodynamischen Vorgängen im Kraftwerksbereich (Wehrfelder, Schleusen)



Wehrsteuerung in HydroAS

Entwicklung und Anwendung eines Wehrsteuerungsmoduls zur Abbildung der Kraftwerkssteuerung an der österreichischen Donau

Achim Naderer und Johannes Rohde, HydroAS Anwendertreffen 2022