Bode Documentation

Release 1

Klaudia Horvath

Jan 16, 2018

Contents

This is a set of Matlab files to model open water channels for control purposes.

CHAPTER 1

IDZ

The IDZ model is implemented based on the paper of Litrico, X., and V. Fromion. “Simplified modeling of irrigation canals for controller design.” Journal of irrigation and drainage engineering 130.5 (2004): 373-383.

1.1 Testing

It is tested using the values of the above mentioned article. It was able to reproduce the presented numbers. The numbers form the article are saved in the excel file, and the file ReadDataLitrico.m does the testing.

1.2 How to use

It is a Matlab function, that can be called with the following code:

```
[p11_inf_hat, p12_inf_hat, p21_inf_hat, p22_inf_hat, au_hat, ad_hat, tu_hat, td_hat,␣
↪yn, x2]=IdzFun(q,n,B,m,Sb,Y0,L)
```

where n is the Manning’s coefficient, q is the discharge, B is the bottom width, Sb is the bottom slope, Y0 is the downstream water depth and L is the length of the channel. The last two outputs are the normal depth and the location where the channel changes between normal flow and backwater. If the whole channel is under backwater this number is zero.

1.3 Implementation

The implementation follows the paper. There is one small thing to notice:

It is slightly different from the paper in order to be able to reproduce the given numbers. This only has effect when the whole canal is under backwater and it is trapezoidal. For the downstream approximation gamma is calculated two ways: taking into account the side slope of the trapezoid and not. (For the upstream it is not relevant, as the water depth does not change.) In order to calculate the frequency approximation the whole formula was used for gamma, and in order to calculate the backwater area the first term (related to the derivative of the top width) was not taken into account.

1.4 Distributed version

CHAPTER 2

Bode plot of the Saint-Venant equations

This function is implemented based on Litrico, Xavier, and Vincent Fromion. “Frequency modeling of open-channel flow.” Journal of Hydraulic Engineering 130.8 (2004): 806-815.

2.1 How to use

It is a Matlab function, that can be called with the following code:

```
[freq, p21_mag, p21_phase_corr, p22_mag, p22_phase_corr, p11_mag, p11_phase_corr, p12_
↪mag, p12_phase_corr ]=...
Bode_fun(m, B, Sb, n,q, x, Y0, min_frequency, max_frequency, TimePoints,SpacePoints)
```

where n is the Manning’s coefficient, q is the discharge, B is the bottom width, Sb is the bottom slope, Y0 is the downstream water depth and L is the length of the channel. The min_frequency and max_frequency are the exponents of 10. If zero is given to any of these quantities, then the minimun/maximum is taken 10 times less/more than the resonance frequency.

The TimePoints are the number of discretisation points in time. If zero is given, a default value of 1000 is used. Space points are the number of discretisation points in space. If zero is given, a default value of space step of 5m is used.

The first output is the frequency in rad/s. The gain values are output in decibels and the phase values in degrees.

2.2 Implementation

Here some more comments should come…

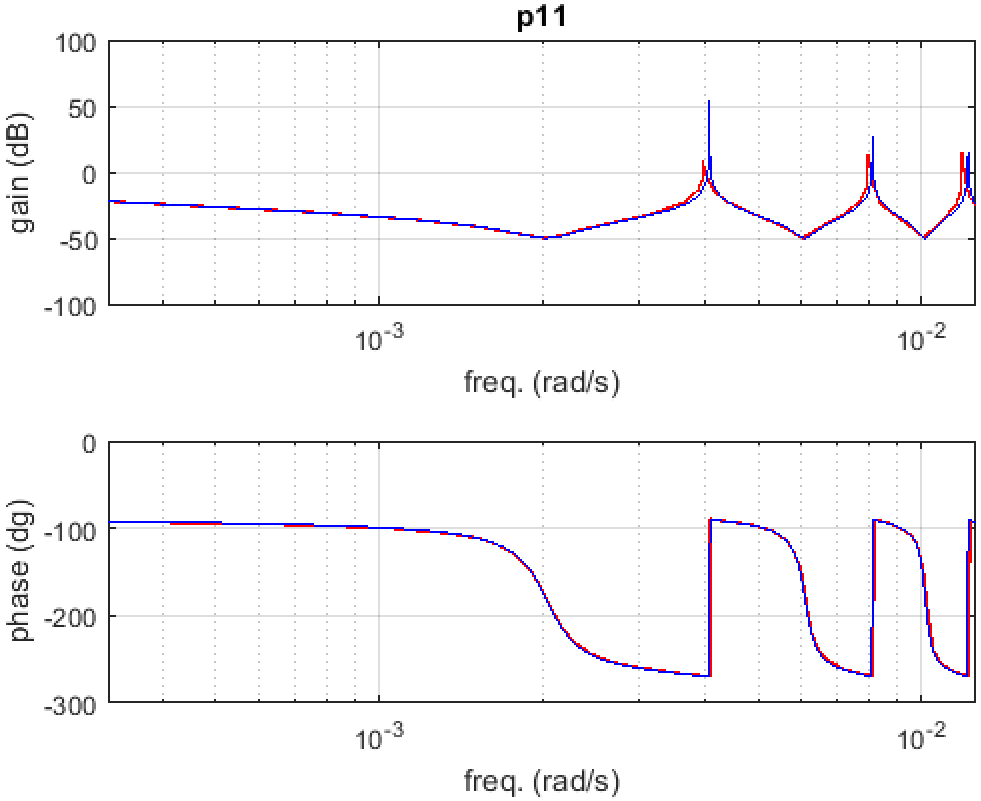
For the calculation of the phase the *atan2* function is used, giving the following results:

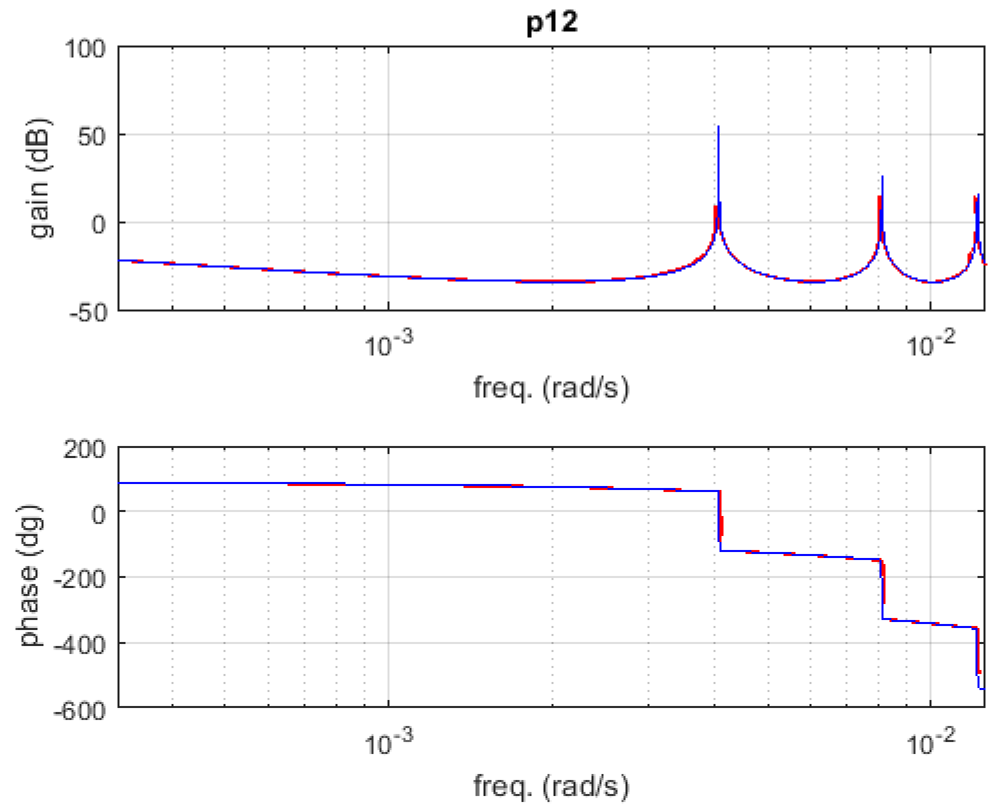
$$\varphi = \arg(z) = \begin{cases} \arctan(\frac{y}{x}) & \text{if } x > 0 \\ \arctan(\frac{y}{x}) + \pi & \text{if } x < 0 \text{ and } y \geq 0 \\ \arctan(\frac{y}{x}) - \pi & \text{if } x < 0 \text{ and } y < 0 \\ \frac{\pi}{2} & \text{if } x = 0 \text{ and } y > 0 \\ -\frac{\pi}{2} & \text{if } x = 0 \text{ and } y < 0 \\ \text{indeterminate} & \text{if } x = 0 \text{ and } y = 0. \end{cases}$$

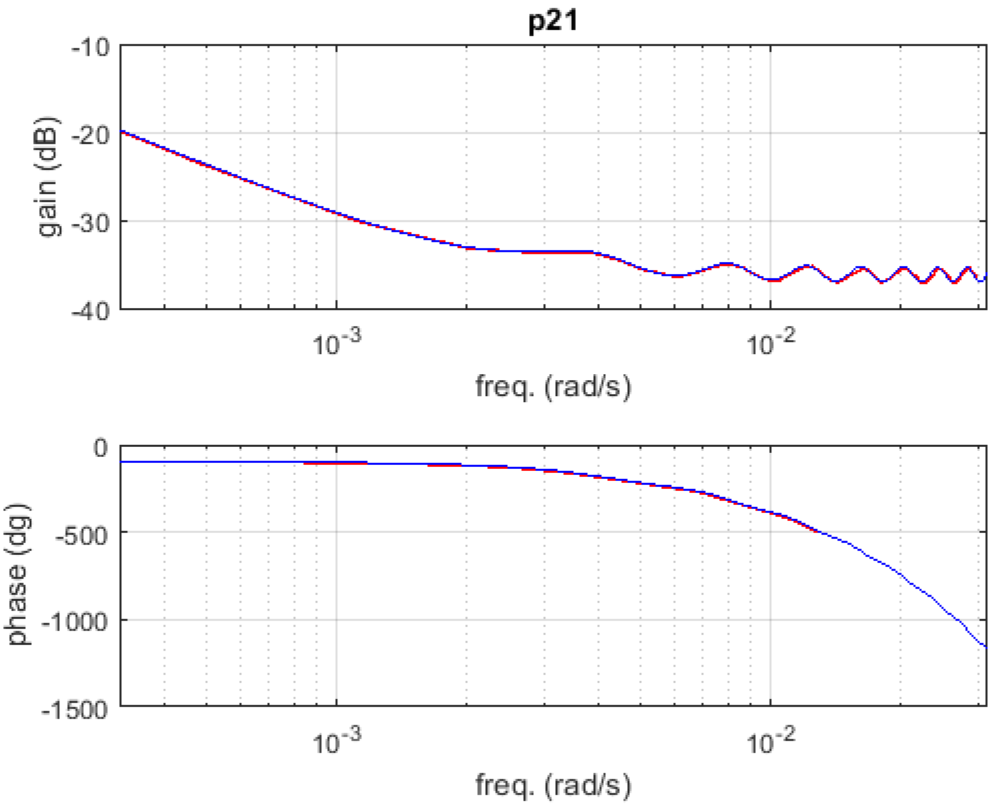
After this operation in some cases 360 degrees are added in order to avoid big jumps. The necessity of adding it or subtracting it depends on the scale. This can be changed by changing the variable *threshold_difference*.

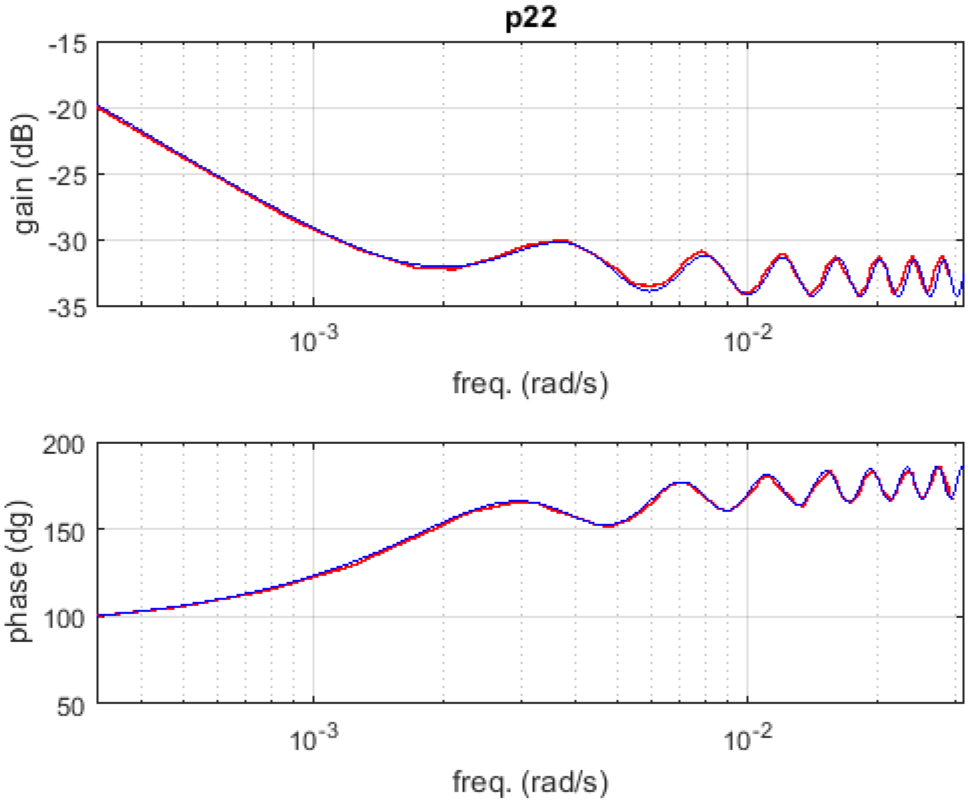
2.3 Testing

The results are compared to the plots in the book Litrico, Xavier, and Vincent Fromion. Modeling and control of hydrosystems. Springer Science & Business Media, 2009. (page 49., Ex. 3.1 for p11 and p21 and p. 69, Ex 3.7 for p21 and p22) Red colour is the literature, blue colour is the written function.




p12
100
50
0
-50
gain (dB)
10^{-3}
10^{-2}
freq. (rad/s)
200
0
-200
-400
-600
phase (dg)
10^{-3}
10^{-2}
freq. (rad/s)


p21
-10
-20
-30
-40
gain (dB)
10^{-3}
10^{-2}
freq. (rad/s)
0
-500
-1000
-1500
phase (dg)
10^{-3}
10^{-2}
freq. (rad/s)



A small note for the tests: for the p11 case the height of the peaks depends on the resolution. However, in practice they are abscissas because the channel is horizontal and frictionless, thus theoretically a there is no force to dissipate the wave.

CHAPTER 3

Steady state profile calculation

3.1 How to use

It is a Matlab function, that can be called with the following code:

```
HinitH= SteadyState(L, z, B ,n, Sb ,q ,h, NodeNum, friction_coef)
```

where n is the Manning's coefficient, q is the discharge, B is the bottom width, Sb is the bottom slope, Y0 is the downstream water depth and L is the length of the channel. *NodeNum* is the number of discretisation points. The last parameter is not compulsory, the default is Chézy. If a zero is given, than it means Manning's roughness is used(and 1 means Chezy).

3.2 Implementation

Integration by Euler.

3.3 Validation

3.3.1 Manning

If the normal depth is calculated by Manning coefficient, then the steady state profile goes to that as well if in the calculation Manning coefficient is chosen. The steady state using Manning is going to normal depth.

3.3.2 Chézy

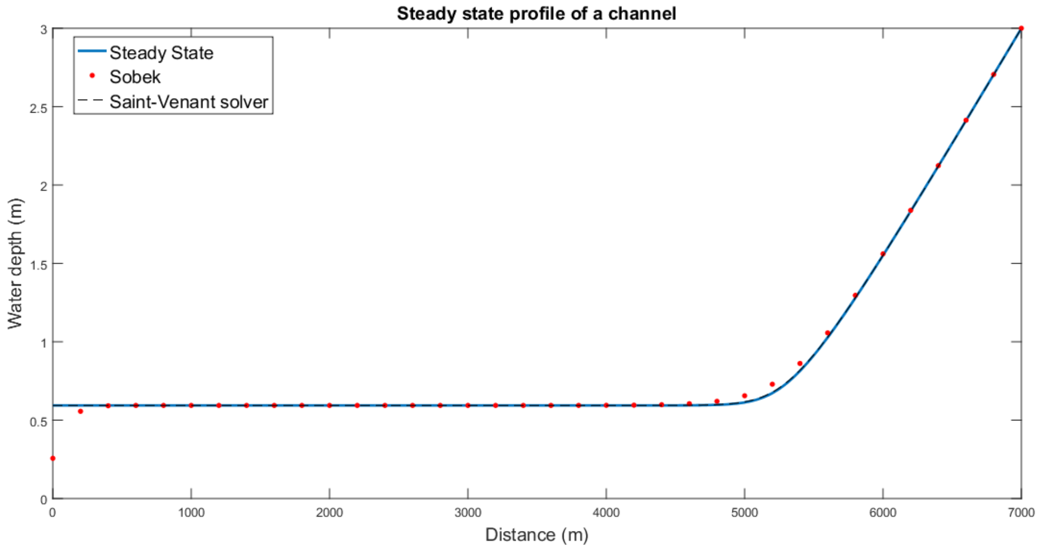
If the Chezy coefficient is chosen, the normal depth calculated by the program will be slightly different than the one calculated by Manning. The advantage of using Chézy coefficient is that the unsteady solver is also using that, so it is easy for initialisation purposes.

3.4 Comparison with Sobek

The steady state profile calculated by Sobek is compared to the one calculated by the steady state and also the result of the SV solver. The following data was used: s=0.0015; L=7000; Bw=3; z=0; n=0.03; q=1.5; h=3;

For the SV: dt=10 NodeNum=1000

The Sobek results are obtained with 200 m spatial discretisation and 5 min timestep, and 33.333 Chézy value. Note that the first two values of the Sobek results are different due to the location of water level and velocity nodes..



CHAPTER 4

Saint-Venant solver

4.1 How to use

It is important to use small enough space step. In the validation example 7 m was used. The SV model works with Chézy friction. It accepts Manning as input, and then it converts it to Chézy. From this constant Chézy value the friction coefficient *Cf* is calculated at every timestep. If given initial profile is used, it should also be coming from a profile calculated with Chézy.

4.1.1 Example

An example:

```
%Input variables

SimTime=13500; %in seconds
s=0.0015;
L=7000;
Bw=3;
z=0;
n=0.03;
dt=10;
SimLength=floor(SimTime/dt)+1;
qu=ones(1,SimLength+1)*1.5;
qd=ones(1,SimLength+1)*1.5;
qu(2:floor(3600/dt))=3;
q=1.5;
h=3;
NodeNum=1000;
HinitGiven=0;
UseInitialization=0;
```

```
[ Time,Hm2 , ~,~] = FiniteVoumeModel11bck(L, z,Bw,n,s,SimTime,dt,qu,qd,h,NodeNum,
↪UseInitialization, HinitGiven );
 plot(Time,Hm2(:,end),'--r')
```

4.1.2 Initialisation module

An initialisation module can be called:

```
[ Time,Hm2 , Hm,HinitH] = InitialiseFiniteVolume(L, z,Bw,n,s,SimTime,dt,qu,qd,h,
↪NodeNum )
```

When steady state initialisation is used, sometimes there is a very little bump /valley due to the different friction treatment. This function runs loops, so that the starting water depth is really the desired one.

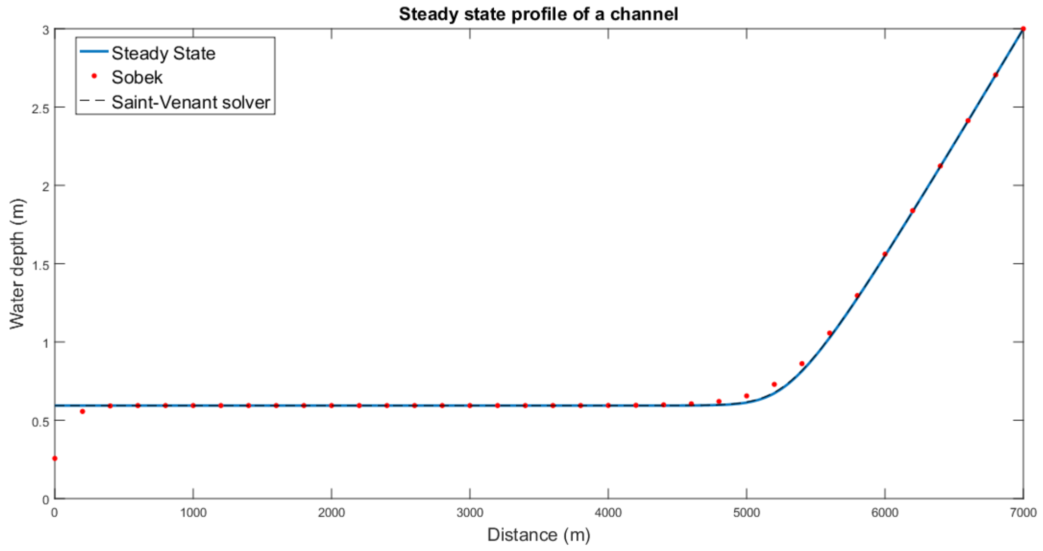
4.2 Validation

4.2.1 Steady state validation

The steady state profile calculated by Sobek is compared to the one calculated by the steady state and also the result of the SV solver. The following data was used: s=0.0015; L=7000; Bw=3; z=0; n=0.03; q=1.5; h=3;

For the SV: dt=10 NodeNum=1000

The Sobek results are obtained with 200 m spatial discretisation and 5 min timestep, and 33.333 Chézy value. Note that the first two values of the Sobek results are different due to the location of water level and velocity nodes.. Note for the Sobek comparison that the value of the first two nodes is a bit different. No similar issue is noticed downstream. Another practical note: if you copy a Sobek model, usually you just need the .dsproj file. However, when a "hot start" is used, then the initialisation file is needed. It is a zip file, located in the dsproj folder.

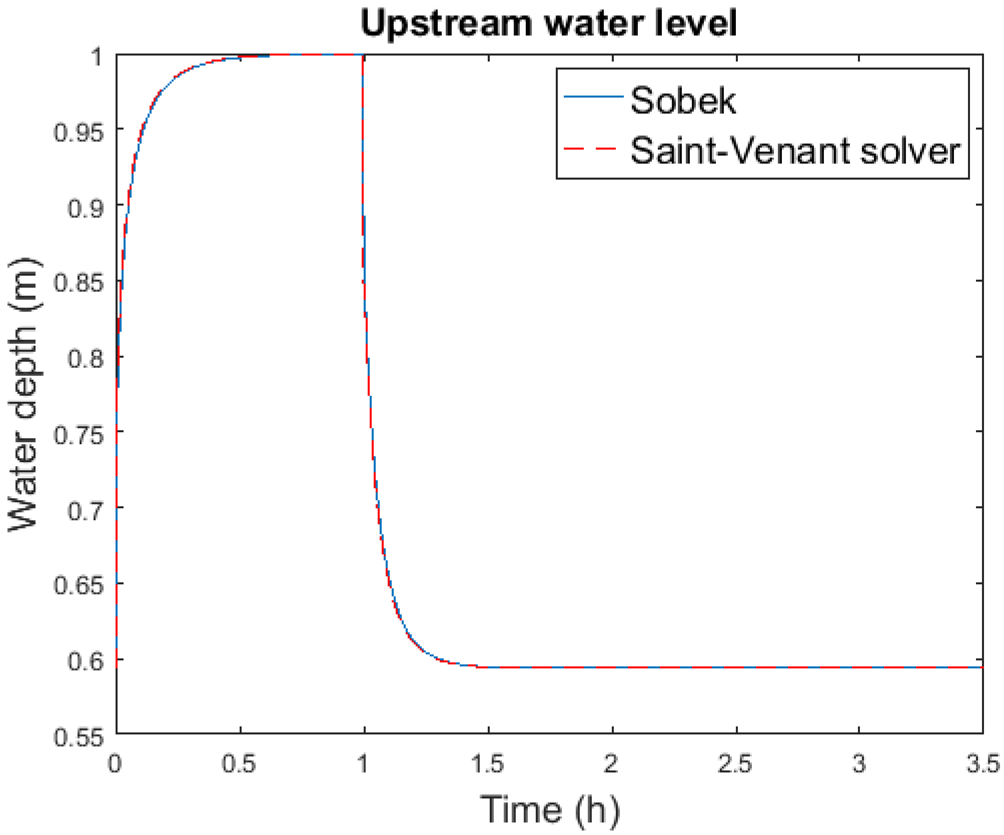


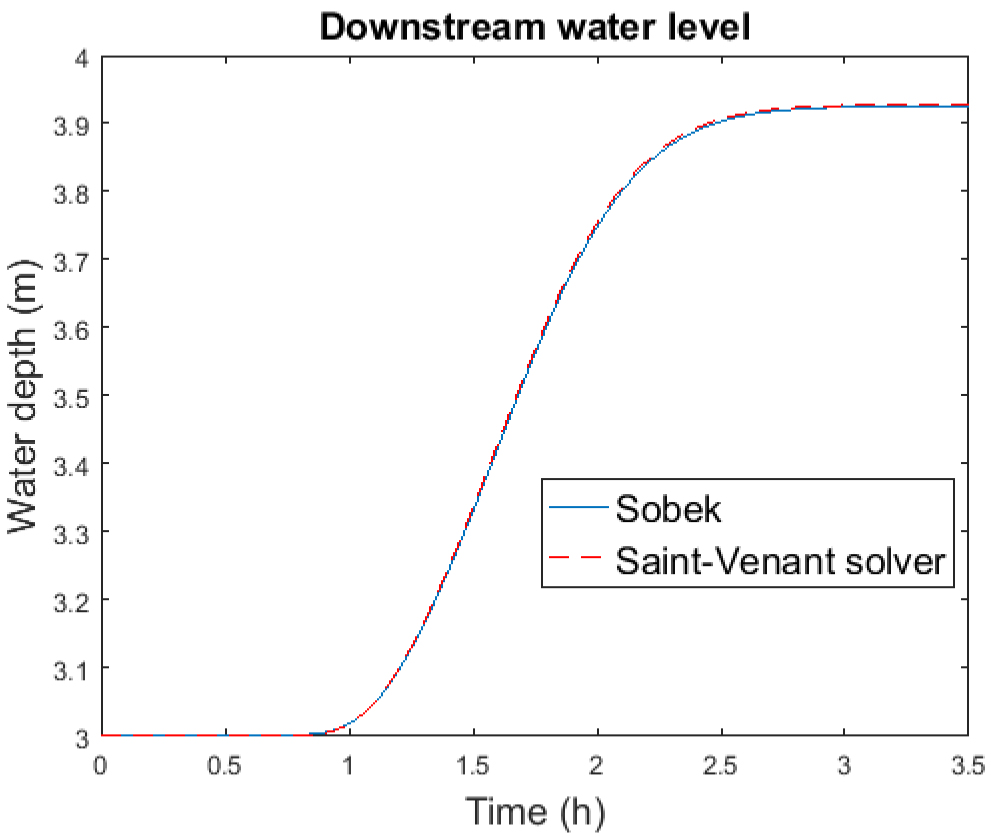
4.2.2 Unsteady state validation

A step response is compared. The following data is used:

SimTime=13500; s=0.0015; L=7000; Bw=3; z=0; n=0.03; dt=10; q=1.5; h=3; NodeNum=1000; HinitGiven=0; UseInitialization=0;

The upstream step is 1h long and the discharge increases to 3 m3/s. The same discretisation is used for Sobek. For the comparison for the upstream the second node is used.





4.3 Implementation

Explicit.

4.4 To do

- The downstream water level (or even structure) boundary condition should be checked.
- Revise the comments
- Revise the boundary conditions
- Document the trapezoidal validation

CHAPTER 5

Calculation of the backwater area

5.1 How to use

This file calculates the backwater area for any point in a channel.

5.1.1 Example

An example:

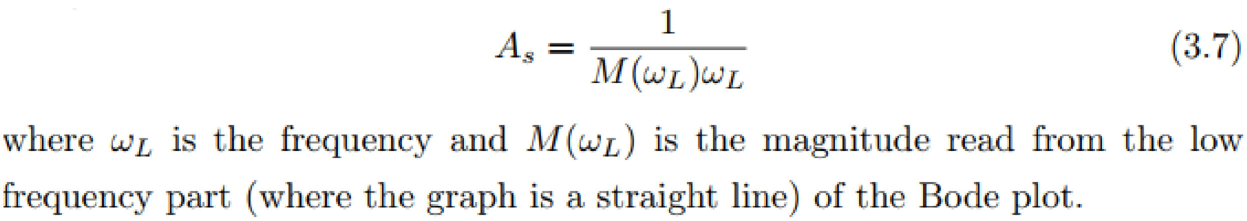
```
[freq, p21_mag_dist, p22_mag_dist, p11_mag_dist, p12_mag_dist, appra11, appra12,␣
↪appra21, appra22 ]=BackwaterAreaApproximation(m, B, Sb, n,q, x, Y0, Startexp,␣
↪Finexp, TimePoints,SpacePoints);
```

5.2 Explanation

There are two files, BackwaterApproximation and BackwaterApproximationFast. The difference is only one line, the fast is not computing all the frequency range, it only computes the lowest frequency.

5.3 Implementation

The implementation uses the formula (Eq. 3.7 from Horváth, Klaudia. "Model predictive control of resonance sensitive irrigation canals." (2013).)



$$A_s = \frac{1}{M(\omega_L)\omega_L} \tag{3.7}$$

where ω_L is the frequency and $M(\omega_L)$ is the magnitude read from the low frequency part (where the graph is a straight line) of the Bode plot.

CHAPTER 6

Notes

6.1 To do

6.1.1 Notes about files

The FiniteVoume11bck is a newer version of the FiniteVoume11. The new version has a different initialization, and creates a file called startparam.mat.

IDZFunDist is used by the file model comparison. (f)

6.1.2 Notes about SV

To fix

How to determine the simulation length for hte initialisation???

Initialisation

what happens when UseInitialisation is zero?

Describe the ini file

Check also for the files for Jeroen the system size issues, so that it works for one size.

Group the functions to a separate folder. Make clear about all files what they do.

Only the file b_channeltype11 uses the FiniteVoume11 file, all the others are using the bck file. The differences should be analyzed. Note that the finitevoume11 file, before I start messing up it today, was modified 05/24/2017. While the back was modified 07/03/2017

%UseInitialization - it is zero if we ininitalize % - it is 1 when we use an initialization % - it is 2 when we create one, it is an initialisation % run % - if it is 3, we load a file with initial values that we % created when it was less than 3

0: an initialisation based on steady state profile is calculated 1: use given time series for initialisation 2: an initialisation based on steady state profile is calculated, H downstream 3: loading created parameters

```
if UseInitialization==2 DischargeDownstreamBoundary=0;
else DischargeDownstreamBoundary=1;
end
if UseInitialization<3
    if UseInitialization==1 H=HinitGiven; HinitH=HinitGiven;
    else (0 or 2) H=HinitH;
    end
else load startparam HinitH=1;
end
if UseInitialization<3 save startparam Qt Ksi Hf Q Uf H Af ch end
```

First, establish the equivalence between the two files. The bck with zero should be the same as the 11 without bck.

Compatiblity

If anywhere you find a Voume11 wihtout bck, it is the same, if bck has the two last parameters zero. Take into account that the new has by default 1 prof factor.

6.1.3 Quotation from Litrico's book

The backwater area $\mathrm{A_d}$ gives the magnitude of the frequency respose of hte sytem for low frequencies, and the time delay τ_d is essential to idnetify the limits of performance of hte system, i.e. the maximum achievalbe frequency ω_c where perturbations can be rejected. This freqency can be estimated by $\omega_c \approx \frac{1}{\tau_d}$ (Aström 2000; Litrico and Fromion 2001). This is hte best achievable real-time performnace of the contolled system; it is not possible to reject perturbations of freqency higher than ω_c with distand downstream feedback control. This is a structural limitation, because it applies for any linear controller.

6.1.4 Backwater todo

- Check if the areas are linearly related and if not try to figure out… or leave it. But.
- make the program faster by setting the right frequecny. One would not be enough???
- compare it to inertial
- Make the file f_ModelComparisonArticle ok. Include the simple bw in it. Make more examples
- Check and finalize the ID area and the IDZ parameter calculation functions
- idz distributed
- analyze the bode plots of trapezoidal and rectangular
- test the approximations on trapezoidal and rectangular
- Describe alpha, kappa, gamma
- Document getting Bode information for the middle and test
- Make a separate backwater area calculation, and call it from all the functions
- write nicely the backwater area computation
- make the two examples to show what it can and what it cannot, use also SV computation
- if you have a lot of time: compare SV chezy and manning with the step. Does the final value depend on those?

- try to make thus the SV manning.
- after having all the files, it would be nice if you could make a complete comparison using csv input
- write IR clearly down, implement the shift

6.1.5 General todo

- finish the model function documentation
- Try to make the old simhydro files run
- Especially make a simhydro folder
- get back the ability for Sobek scripting

6.1.6 Joris paper todo

- Make a study and understanding about upstream bodes
- Make a connection between categories and normal flow
- Revise linear inertial
- Based on this, review Hayami, also how distributed they are
- Based on this, review id, also how distributed they are
- Based on this, review idr, also how distributed they are
- code the IR found shift
- check linear inertial model
- make ir distributed, document it

6.2 For backwater article

6.2.1 Article content

Goal: distributed backwater approximations

Show three step responds and its reproducibility We are going to show / test different methods: - based on IDZ - based on Bode - nothing - compared to linear inertial model

Make comparisons in time and space domain, using different examples Show results with fricitonless flat canal

How to approach the middle of the reach?

6.2.2 Dealing with IDZ

$$\frac{\partial q}{\partial t} + 2V_0(x)\frac{\partial q}{\partial x} - \beta_0(x)q + (C_0(x)^2 - V_0(x)^2)T_0(x)\frac{\partial y}{\partial x} - \gamma_0(x)y = 0,$$

- For the delay it is clear.

- For the backwater area I found a method that should be tested.
- For the zeros?

6.3 To do Joris paper

- order and check the files
- make closed loop runs (Python or Matlab?)
- make the bode plots - again!
- Explain a ctr talk about peak height with bode plots, talking about the sampling time
- Make a test with PID as well (start from typical tuning rules)
- Write / search for my MPC algorithm

1. Use the categories of Joris, add IR
2. Use the cat. of Joris, make freq. plot
3. Think on Inertial how to do it
4. Put the closed loop - RTC tools???
5. Look for how to make the Bode plot analytically
6. Create bode from time

6.4 Publication plans

Publication bw area, reference Pau, reference Fatiha

6.4.1 To read

Yolanda Bolea, Vicenç Puig, and Antoni Grau. Discussion on muskingum versus integrator-delay models for control objectives.
Journal of Applied Mathematics, 2014, 2014.

CHAPTER 7

IDZ Distributed version

The distributed version of the IDZ model is based on Litrico, Xavier, and Vincent Fromion. “Analytical approximation of open-channel flow for controller design.” Applied Mathematical Modelling 28.7 (2004): 677-695.

7.1 Testing

7.2 How to use

It is a Matlab function, that can be called with the following code:

```
[tuM,tdM,pum,pdm,puma]=IdzFunDist(q,n,B,m,Sb,Y0,L, NodeNumGiven)
```

where n is the Manning’s coefficient, q is the discharge, B is the bottom width, Sb is the bottom slope, Y0 is the downstream water depth and L is the length of the channel. NodeNumGiven is the number of spatial discreatisation nodes. The last optional argument is used to change the space discretization. If it is 1, then the same discreatization as for the SV is used.

7.3 Implementation

Expressions are derived for backwater area, zeros and the delay for any given point in the reach based on Litrico, Xavier, and Vincent Fromion. “Analytical approximation of open-channel flow for controller design.” Applied Mathematical Modelling 28.7 (2004): 677-695.

CHAPTER 8

Integrator resonance model

The IR model is implemented based on the paper of Van Overloop, P. J., et al. “Identification of resonance waves in open water channels.” Control Engineering Practice 18.8 (2010): 863-872.

8.1 Testing

8.2 How to use

It is a Matlab function, that can be called with the following code:

```
[num,den,sysidr,Mr_calc,Om_r_calc, num2, Om_r_calc2]=ident_idr_new(b0 ,h , m ,q ,x,n,
↪Mr,Om_r,UseTravelTime)
```

where n is the Manning’s coefficient, q is the discharge, B is the bottom width, Sb is the bottom slope, Y0 is the downstream water depth and L is the length of the channel.

8.3 Implementation

8.4 Background

The IR model is deduced from the linarized Saint-Venant equations. It is a second order wave with an integrator. Thus, it has three free parameters: - integrator gain (backwater area) - frequency (inverse of a wave travel time) - damping

The first parameter determines its behaviour reaching steady state. For open water channels this can be approximated with the backwater area. Changing this parameter would change the final steady state (and hence the mass balance). The second parameter is the frequency. This has also very clear physical meaning, and it can be very well approximated by using the travel time of a wave. The third parameter, the damping is the most difficult to approximate, because the formula (1/(2*Om_r*A*Mr)) contains the resonance peak. This can be approximated by formulas, but it is not correct.

For the CF reach, the problem is that the wave is not “high” enough. It can be higher by decreasing the damping. However, even using very small damping (1e-50), the wave height hardly increases.

Thus, there is not much possibility to tune this structure more. The first two parameters are set to the best (the backwater area is good, that can be seen in the realistic simulation), and now also the resonance frequency is well calculated. (I do suggest to use the formula instead of tuning). The Mr can be a bit changed, but the situation will hardly improve.

8.4.1 Reasoning

The IR is not meant to be a model for time domain simulations. It is meant to be for control purposes. And also, sometimes I am not convinced how good it is. In time domain, it will never be better as our results now. The question is what is your final goal.

CHAPTER 9

Linear inertial model

The linear inertial model is the linearized version of the inertial wave equation. That is the Sainv-Venant equations without the convective term. The equation is described (including discertization) for example in Montero, R. A., Schwanenberg, D., Hatz, M., & Brinkmann, M. (2013). Simplified hydraulic modelling in model predictive control of flood mitigation measures along rivers. Journal of Applied Water Engineering and Research, 1(1), 17-27.

9.1 How to use

It is a Matlab function, that can be called with the following code:

```
[x A Bd D,Hinit]=LinearInertial(z,b,h0,n,q0,s,l,dt,NodeNumGiven,ModelType)
```

where n is the Manning's coefficient, q is the discharge, B is the bottom width, Sb is the bottom slope, Y0 is the downstream water depth and L is the length of the channel. dt is the discretization time and NodeNumGiven is the number of space nodes used for discretization. ModelType has not function for the time being.

9.2 Example

See the file TestLineraInertial.m

CHAPTER 10

Contribute

- Source Code: github.com/Klaudia/simple-models

CHAPTER 11

Support

If you are having issues, please let us know. We have a mailing list located at: hklau85@gmail.com

CHAPTER 12

License

The project is licensed under the BSD license.