

# Memo

<b>Datum</b> 17 december 2018	<b>Aantal pagina's</b> 7
<b>Contactpersoon</b> Menno de Ridder	<b>E-mail</b> Menno.deRidder@deltares.nl
<b>Onderwerp</b> Python HBdataMonitor	

---

## 1 Introductie

De data van de hydraulische belastingen zijn opgeslagen in SQLite databases. Dit heeft als voordeel dat de data op een compacte manier is opgeslagen. Echter, het is hierdoor wel lastiger om de data te controleren of inzichtelijk te maken voor de beheerders van deze data (verder in deze memo kortweg 'beheerders' genoemd). De data is namelijk niet in één ('platte') tabel opgeslagen, maar in een verzameling samenhangende tabellen en kan alleen met SLQL queries verkregen worden. Om de inhoud van de databases toch te kunnen weergeven is een Python tool gemaakt die deze queries uitvoert op basis van een standaard invoer. Met behulp van deze tool kan de data voor een gegeven conditie worden weggeschreven en grafisch weergegeven.

De tool is gemaakt voor de beheerders en ontwikkelaars van de hydraulische databases. Dit is een kleine gebruikersgroep, die bovendien flexibel wil zijn in mogelijkheden voor aanpassingen van de gebruiksopties. Daarom is geen user interface gemaakt, maar de functies zijn beschikbaar via een Python package. De gebruiker kan deze package installeren en daarna zelf in Python de functies aanroepen.

## 2 Werking HBdataMonitor

De HBdataMonitor is geschreven in de opensource programmeertaal Python 3.6. Dit betekent dat de HBdataMonitor geen additionele software vereist om de tool te gebruiken. Wel zijn een aantal Python packages nodig voordat de HBdataMonitor gebruikt kan worden. Deze packages worden meestal standaard geïnstalleerd met de Python distribution en anders zijn de packages beschikbaar via PIP (<https://pypi.org/project/pip/>).

De Python tool is gegeven door een Python-class met meerdere functies. Voor de gebruiker zijn drie functies van belang, waarmee de informatie van een database kan worden weergegeven, de data uit de database gehaald kan worden en een figuur van de data gemaakt kan worden. Een overzicht van alle functies is gegeven in hoofdstuk 4. Zie het voorbeeld voor de werking van de HBdataMonitor (hoofdstuk 5). Verder is de tool zo gemaakt dat het voor elk watersysteem gebruikt kan worden. Dus zolang de structuur van de database het zelfde blijft kan de tool gebruikt worden.

De volgende toepassingen zijn mogelijk met de tool:

1. Weergeven van de database informatie en aanwezige stochasten.
2. Wegschrijven van de invoerparameters met alle beschikbare waarden.
3. Wegschrijven van een uitvoerparameter gegeven een reeks van invoerparameters. Dit kan voor een onbeperkt combinatie van invoerparameters. Ook de databases kunnen als invoer worden gebruikt, zodat verschillende versies van databases vergeleken kunnen worden. Voor de invoerparameters waarvoor meer dan 1 waarde is gewenst, moet de gewenste reeks waarden expliciet worden opgegeven, maar ook kan simpelweg met 'all' worden aangegeven dat alle beschikbare waarden getoond moeten worden.
4. Plotten van de parameters, waarbij tot maximaal vier dimensies kunnen worden meegeven. De x-as, y-as, z1-as (kleuren), z2-as (symbolen). De waarden en volgorde op de assen kunnen ook worden aangepast door de reeks van de invoerparameters te wijzingen.

### 3 Input voorbeeld

De werking van het script is weergeven in onderstaand voorbeeld voor een database van het IJsselmeer. In dit voorbeeld wordt stap voor stap de invoer uitgelegd.

#### 3.1 Download the HBdataMonitor

De HBdataMonitor is als Python package beschikbaar via PIP. De HBdataMonitor kan worden geïnstalleerd door het volgende commando in de Python terminal uit te voeren:

```
pip install HBdataMonitor
```

De Beheertool wordt dan gedownload en geïnstalleerd. Als de tool eenmaal is geïnstalleerd kan het worden geïmporteerd in een Python script. Ook kan de package handmatig worden geïnstalleerd vanaf <https://pypi.org/project/HBdataMonitor/>.

#### 3.2 Importeren van de HBdataMonitor

De HBdataMonitor is geschreven in een Python class. Om deze Python class te kunnen gebruiken moet deze eerst worden geïmporteerd. Met de volgende code kan de HBdataMonitor worden geïmporteerd (let op de hoofdletters):

```
from HBdataMonitor import HBdataMonitor
```

De HBdataMonitor kan dan worden aangeroepen met de volgende regel:

```
## open inspectietoolHB class  
HBtool = HBdataMonitor()
```

#### 3.3 Algemene informatie database

Om de algemene informatie van de database weer te geven kan de functie `get_DB_info()` gebruikt worden. Deze functie heeft als input de locatie (combinatie van pad- en

bestandsnaam) van de database. Als uitvoer geeft het de algemene informatie weer in het scherm en schrijft het de invoerparameters met alle waarden naar een ascii-file.

```
myDatabase = 'd:\InspectietoolHB\data\GR2017_IJsselmeer_6-1_v01.sqlite'
## show general info - input variables are written to file
HBtool.get_DB_info(myDatabase)
```

Met behulp van deze functie kan de gebruiker de aanwezige invoerparameters van de database verkrijgen. Daarnaast kunnen de waarden van deze invoerparameters worden opgezocht in de ascii-file. Deze informatie is nodig voor de volgende stap, waar de invoerparameters moeten worden gespecificeerd om de data uit de database te halen. Voor de database van het IJsselmeer ziet het resultaat er als volgt uit:

```
GeneralId: 7
NameRegion: IJssel lake
TrackID: 7049
Track: 49
CoordinateSystem: EPSG: 28992
CreationDate: 2016-12-06 18:29
Number of input Variables=5
WINDS
MEERP
WindDir
loc
ClosingSituation
Number of result Variables=5
ZMAX
Hsign
Tpiek
Dir
Tm-1,0
```

In de ascii-file (Variables.txt) worden de beschikbare invoerparameters en hun waarden weergegeven. Dit wordt gedaan door per regel de waarden voor de invoerparameter weer te geven (zie onderstaande voorbeeld).

---

```
InputVariables
WINDS;0.0;14.0;19.0;22.0;25.0;28.0;31.0;34.0;38.0;42.0;
MEERP;-0.4;-0.1;0.4;1.0;1.8;
WindDir;225.0;22.5;45.0;67.5;90.0;112.5;135.0;157.5;180.0;202.5;247.5;270.0;292.5;315.0;337.5;360.0;
loc;YM_2_6-1_dk_01009;YM_1_6-1_dk_01000;YM_2_6-1_dk_00997;YM_1_6-1_dk_00987;
ClosingSituation;1;
```

---

### 3.4 Data uit de database halen

Om de data uit de database te krijgen moet worden aangegeven voor welke waarden van de invoerparameter de uitvoerparameter geselecteerd moet worden. Deze invoerparameters worden meegegeven in een Python dictionary. Deze dictionary bestaat uit de invoerparameters als key en een lijst als value (zie onderstaande voorbeeld). Op deze manier kunnen per invoerparameter de te selecteren waarden worden opgegeven. Dit kan één enkele waarden

zijn of een reeks van waarden. Wanneer “all” wordt gebruikt worden alle beschikbare waarden voor die gegeven invoerparameter in de database gebruikt.

De volgorde van de reeks van waarden in de lijst is ook de volgorde van het plotten in de volgende stap. Dit kan nuttig zijn als bijvoorbeeld als de locatie op x-as moet worden weergegeven voor een bepaalde reeks. Op basis van deze invoerparameters wordt voor elke combinatie de data uit de database gehaald.

Het is wel belangrijk dat voor elke parameter ten minste één waarde wordt meegegeven. De locatie is ook aangegeven als parameter (“loc”) en moet dus ook worden gespecificeerd. Hetzelfde geldt voor de ClosingSituation (“ClosingSituation”) en de locatie van de database (“DB”). Het is dus ook mogelijk om verschillende databaseversies met elkaar te vergelijken, mits de verzameling namen van invoerparameters en uitvoerparameters overeenkomt; het is niet mogelijk om databases van verschillende gebieden met elkaar te vergelijken.

Naast de invoerparameters moet ook een uitvoerparameter worden gedefinieerd. Dit wordt gedaan door de deze uitvoerparameter als invoer mee te geven aan de functie *HBOutput* zoals weergegeven in onderstaand voorbeeld.

```
## input variables
variables = {'WINDS': ['all'], 'MEERP': [-0.4, 1.8], 'WindDir': [90, 180, 270, 360],
'loc': ['YM_2_6-1_dk_00997'], 'ClosingSituation': [1], 'DB': [myDatabase]}
## extract data - data is written to a file
HBtool.HBOutput('Hsign', variables)
```

De data wordt weggeschreven naar een ascii-file (standaard result.txt). Eerst worden onder 'constants' de parameters getoond die constant zijn gehouden. Daarna worden de onder 'varied' de namen van de uitvoerparameter en de variërende invoerparameters getoond, gevolgd door een tabel met de geselecteerde waarden van deze parameters. De eerste kolom van deze tabel is altijd de uitvoerparameter. Voor de combinaties die niet bestaan wordt -999 getoond. Zie onderstaande tekst voor het gegeven voorbeeld.

---

```
constants
loc=YM_2_6-1_dk_00997
ClosingSituation=1
DB=d:\InspectietoolHB\data\GR2017_IJsselmeer_6-1_v01.sqlite
varied
Hsign;WINDS;MEERP;WindDir;
0.0;0.0;-0.4;90;
0.0;0.0;-0.4;180;
0.0;0.0;-0.4;270;
0.0;0.0;-0.4;360;
0.0;0.0;1.8;90;
0.0;0.0;1.8;180;
0.0;0.0;1.8;270;
0.0;0.0;1.8;360;
0.505;14.0;-0.4;90;
0.895;14.0;-0.4;180;
0.97;14.0;-0.4;270;
0.66;14.0;-0.4;360;
```

---

0.736;14.0;1.8;90;  
 1.187;14.0;1.8;180;  
 1.203;14.0;1.8;270;  
 0.862;14.0;1.8;360;  
 0.715;19.0;-0.4;90;  
 1.224;19.0;-0.4;180;  
 1.309;19.0;-0.4;270;  
 0.912;19.0;-0.4;360;  
 1.048;19.0;1.8;90;  
 1.591;19.0;1.8;180;  
 1.593;19.0;1.8;270;  
 1.188;19.0;1.8;360;  
 0.842;22.0;-0.4;90;  
 1.436;22.0;-0.4;180;  
 1.526;22.0;-0.4;270;  
 1.079;22.0;-0.4;360;  
 1.244;22.0;1.8;90;  
 1.849;22.0;1.8;180;  
 1.853;22.0;1.8;270;  
 1.401;22.0;1.8;360;  
 0.955;25.0;-0.4;90;  
 1.641;25.0;-0.4;180;  
 1.707;25.0;-0.4;270;  
 1.246;25.0;-0.4;360;  
 1.43;25.0;1.8;90;  
 2.123;25.0;1.8;180;

---

### 3.5 Weergeven data

Om de data te kunnen weergeven is een plot functie aanwezig. Deze functie leest de ascii-file, weggeschreven in de vorige stap, en maakt er een figuur van. Als invoer kunnen de parameters worden mee gegeven die op de x en y-as worden gezet. Daarnaast is het mogelijk om twee z variabelen mee te geven, waarbij de eerste z variabele de kleur en de tweede z variabele het symbool weergeeft. Hierbij moet worden opgemerkt dat het aantal kolommen in de ascii-file (dus het aantal variërende variabelen) overeen moet komen met de input van de `create_plot()` functie. Zie onderstaand voorbeeld voor de invoer van de `create_plot` functie.

```

## plot results based on the text file
HBtool.create_plot(X='WINDS',Y='Hsign',Z1=WindDir,Z2='MEERP')
  
```

Als er maar 1 of 2 variërende input variabelen (dus 2 of 3 kolommen in de ascii-file) zijn moet de definitie van `Z1` of/en `Z2` worden weggelaten. Het is ook niet mogelijk om een figuur te maken van een ascii-file met meer dan 4 variërende invoerparameters. Standaard wordt er een png en pdf weggeschreven naar de locatie van het script. Het is ook mogelijk om de naam en locatie te wijzingen doormiddel van het keyword `path` (`path='C://figuur'`).

Voor de figuur wordt standaard het rainbow kleuren schema en standaard symbolen gebruikt. Dit kan ook worden aangepast door de class parameters `colors` en `linestyle` te veranderen. Zie onderstaande voorbeeld voor het wijzingen van de kleuren en symbolen. Merk op dat het aantal

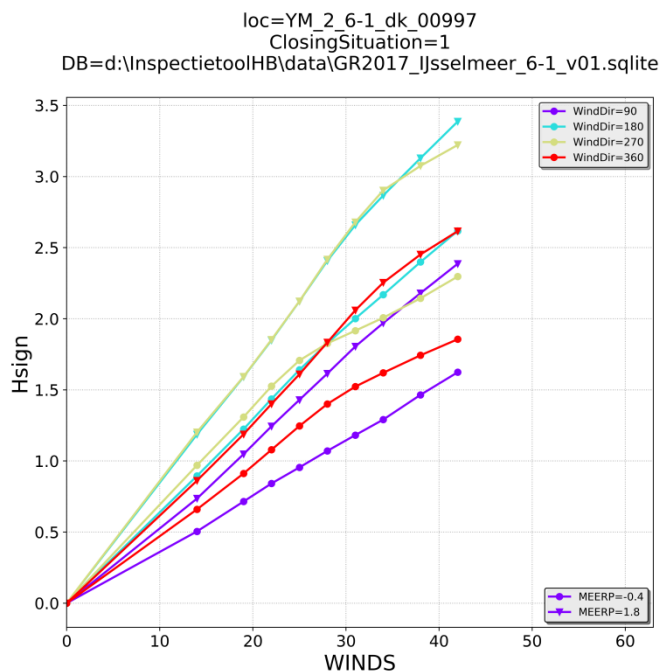
kleuren en symbolen wel groter moet zijn dan het aantal waarden van de betreffende invoerparameter (Z1 of Z2).

```
HB.colors = ['r', 'b']
HB.linestyle = ['o', '<-']
```

### 3.6 Resultaat

De figuur die wordt gemaakt bij dit voorbeeld is getoond in Figuur 3.1. De locaties staan op de x-as en de golfhoogte op de y-as. Met de kleuren zijn de verschillende meerpeilen aangegeven en de twee windsnelheden zijn als symbolen geplot. In de titel van de figuur staan de invoerparameters die constant zijn gehouden.

Figuur 3.1      *Figuur parameters IJsselmeer.*



## 4 Functies

Een overzicht van de functies is gegeven in deze paragraaf, waarbij het resultaat en de invoer van de functie zijn beschreven. Als een functie een variabele teruggeeft is dat aangegeven met return.

#### **read\_sql\_table(path, command)**

path                = locatie van de dataset  
command            = SQLite commando

return: Een Pandas Dataframe met de tabel die bij het SQLite commando hoort.

**read\_variables(path, columnId, inputvar=True)**

path               = locatie van de dataset  
columnId          = columnID van de parameter zoals gegeven in de database  
inputvar          = switch tussen invoerparameter of uitvoerparameter (standaard True)

return: Een gesorteerde array met de waarden van de parameter.

**get\_DB\_info(path, fname='Variables.txt')**

path               = locatie van de dataset  
fname              = naam van de ascii-file (standaard Variables.txt)

Print de informatie in de tabel General van de database.  
Print de invoer en uitvoer parameters.  
Schrijft de waarden van alle invoerparameters naar een ascii-file.

**HBoutput(Y, inputX, fname='result.txt')**

Y                   = uitvoer parameter  
inputX             = dictionary met invoerparameter  
fname               = naam van de ascii-file

Schrijft op basis van de invoer alle resultaten naar een ascii-file.

**create\_plot(Y=0, X=1, Z1=0, Z2=0, fname='result.txt', path='result')**

Y                   = parameter op de y-as  
X                   = parameter op de x-as  
Z1                  = parameter op de Z1-as (kleuren)  
Z2                  = parameter op de Z2-as (symbolen)  
fname               = naam van de ascii-file  
path                = locatie van figuur

Leest de ascii-file in en maakt er een figuur van.