



## Table of Content

<b>Introduction</b>	<b>1</b>
Purpose and Scope	1
Organization	1
Point of Contact	2
Reference	2
<b>Background   Application's Functions</b>	<b>3</b>
Detailed Description of Functions	3
Preparation of function inputs	7
Results of functions (Simulation Output)	8
References	10
<b>Operating Instructions (step-by-step-guide)</b>	<b>11</b>
Where to download the software application and how to get started?	11
Requirements	11
IntroPage   Define Parameter	11
Simulation Page	13

# Introduction

**SensinSilico** is a simulation software published together with the peer-reviewed article [xxxxx](#) in [xxxxx](#). It studies the error propagation when combining two (electrochemical) sensors with different response times. This document is meant as installation and operation manual to use the software and explains relevant application's functions.

## Purpose and Scope

In nature, most biogeochemical processes are based on the interchange of weak acid-base equilibria. Yet, in some of these equilibria, it occurs that one species may be harmful or toxic to the environment while the other species is harmless or even essential to biogeochemical processes. To describe both species as well as the overall behaviour of the acid-base equilibrium, it has become common in environmental analysis to monitor the pH and one of the two parameters of the equilibrium. By obtaining the values of the two, it becomes possible to calculate the third parameter and subsequently the total parameter.

Two examples of acid-base equilibria that are critical to environmental processes are the ammonium/ammonia exchange, expressed as total ammonia-nitrogen (TAN)<sup>1,2</sup>, and the sulfide equilibrium, which includes H<sub>2</sub>S and HS<sup>-</sup> as total dissolved sulfide (TDS)<sup>3,4</sup>:



When two different sensors are required to determine a certain (total) parameter in a heterogeneous system as is done in the above-mentioned examples, sensor alignment and experiment design become critical. Especially in heterogeneous systems where parameters may fluctuate greatly within a few  $\mu\text{m}$ , it must be ensured that the individual sensors are positioned exactly at the same place. However not only the location is crucial, but also timing since parameters can vary strongly over time. Additionally, each sensor has an individual (delayed) response time. To derive individual parameters correctly from the measured ones, the timing between the different sensors must be coordinated to avoid error propagation and falsified results.

Since we know that reality is not always ideal and time pressure can lead to non-ideal measurement conditions, we have developed the software **SensinSilico**. The aim is to simulate the error propagation that can occur when calculating sum parameters from two individual sensors with different response times (e.g. one slow and one fast sensor). While all weak monovalent acid-base equilibria can be simulated in theory, TAN and TDS, as common examples, have been pre-programmed. The simulation assumes a target total parameter concentration and a set of user-defined pH step changes. The target concentration is then compared to the concentration calculated from the simulated data from the two individual sensors with different response times. The error propagation in the calculation procedure associated with the sensor response time mismatch can then be determined within the software.

## Organization

In the next chapter, **Background | Functions of the Application**, we describe the theory required to understand the simulation, how to prepare the input parameters for each function and how to calculate the error propagation software. To illustrate the handling of the software, we have created a step-by-step guide for the TAN equi-

librium. We recommend recapping this example to get a feel for how the software works. If you have difficulties running the software or encounter any bugs, we encourage reaching out to us (cf. **Point of Contact**).

## Point of Contact

In case of encountering any bugs or if you have any further questions, please feel free to reach out to [info@silviazieger.com](mailto:info@silviazieger.com).

## Reference

- (1) Water quality standards: Total Ammonia Nitrogen - Responsible Seafood Advocate. Global Seafood Alliance. <https://www.globalseafood.org/advocate/water-quality-standards-total-ammonia-nitrogen/> (accessed 2023-06-13).
- (2) Mook, W. T.; Chakrabarti, M. H.; Aroua, M. K.; Khan, G. M. A.; Ali, B. S.; Islam, M. S.; Abu Hassan, M. A. Removal of Total Ammonia Nitrogen (TAN), Nitrate and Total Organic Carbon (TOC) from Aquaculture Wastewater Using Electrochemical Technology: A Review. *Desalination* 2012, 285, 1–13. <https://doi.org/10.1016/j.desal.2011.09.029>.
- (3) Steininger, F.; Koren, K.; Revsbech, N. P.; Marzocchi, U. Microsensor for Total Dissolved Sulfide (TDS). *Chemosphere* 2023, 323, 138229. <https://doi.org/10.1016/j.chemosphere.2023.138229>.
- (4) Hydrogen sulfide toxic, but manageable - Responsible Seafood Advocate. Global Seafood Alliance. <https://www.globalseafood.org/advocate/hydrogen-sulfide-toxic-but-manageable/> (accessed 2023-06-13).

# Background | Application's Functions

For the sake of illustration, we focus here on the total ammonia (TAN) example. For simulating total dissolved sulfide (TDS) or other weak monovalent acids, one only has to adjust the respective parameter values according to **Table 2-3**.

It is assumed that pH and one of the two species in the acid-base equilibrium will be monitored over time in this and any similar experiment. The respective other species are calculated based on the **Henderson-Hasselbalch** equilibrium for weak acid/base pairs and the total parameter is the sum of the two individual species at each time point.

For our demonstration, we used values for the sensor properties based on what we could find on sensor manufacturer's websites:

**TABLE 1** Commercially available sensors for pH,  $\text{NH}_4^+$ ,  $\text{NH}_3$  and  $\text{H}_2\text{S}$  and their corresponding response times

Analyte	Type	Response time /s	Supplier
pH	Optical	<60	Pyroscience <sup>1</sup>
	Optical	<120	Presens <sup>2</sup>
	Potentiometric	1-60	Krohne <sup>3</sup>
	Potentiometric	<10	Unisense <sup>4</sup>
	Potentiometric	<45	MetrOhm <sup>5</sup>
$\text{NH}_4^+$	ISE	<180	Hach <sup>6</sup>
$\text{NH}_3$	Severinghaus-type	<60	Fisher <sup>7</sup>
	Severinghaus-type	<600	MetrOhm <sup>8</sup>
$\text{H}_2\text{S}$	Amperometric	<10	Unisense <sup>9</sup>
	Amperometric	<25	Sulfilogger <sup>10</sup>

## Detailed Description of Functions

Since we focus on pH-dependent two-sensor systems, we start our simulation with the pH value and its fluctuation over time, which is set by the user. As a target concentration, we simulate a step function assuming an infinite fast sensor response. For the example, we assume pH fluctuations of 4, 12.4, and 8.3 with a plateau time of 75 seconds. Furthermore, we assume that the system under investigation is based on a weak acid/base equilibrium, allowing us to use the **Henderson-Hasselbalch** equation to determine the analyte concentrations.

### Henderson Hasselbalch

$$\text{pH} = \text{pK}_a + \log \frac{[\text{Base}]}{[\text{Acid}]} \quad \text{and} \quad [\text{Total}] = [\text{Base}] + [\text{Acid}] \quad (1) \text{ and } (2)$$

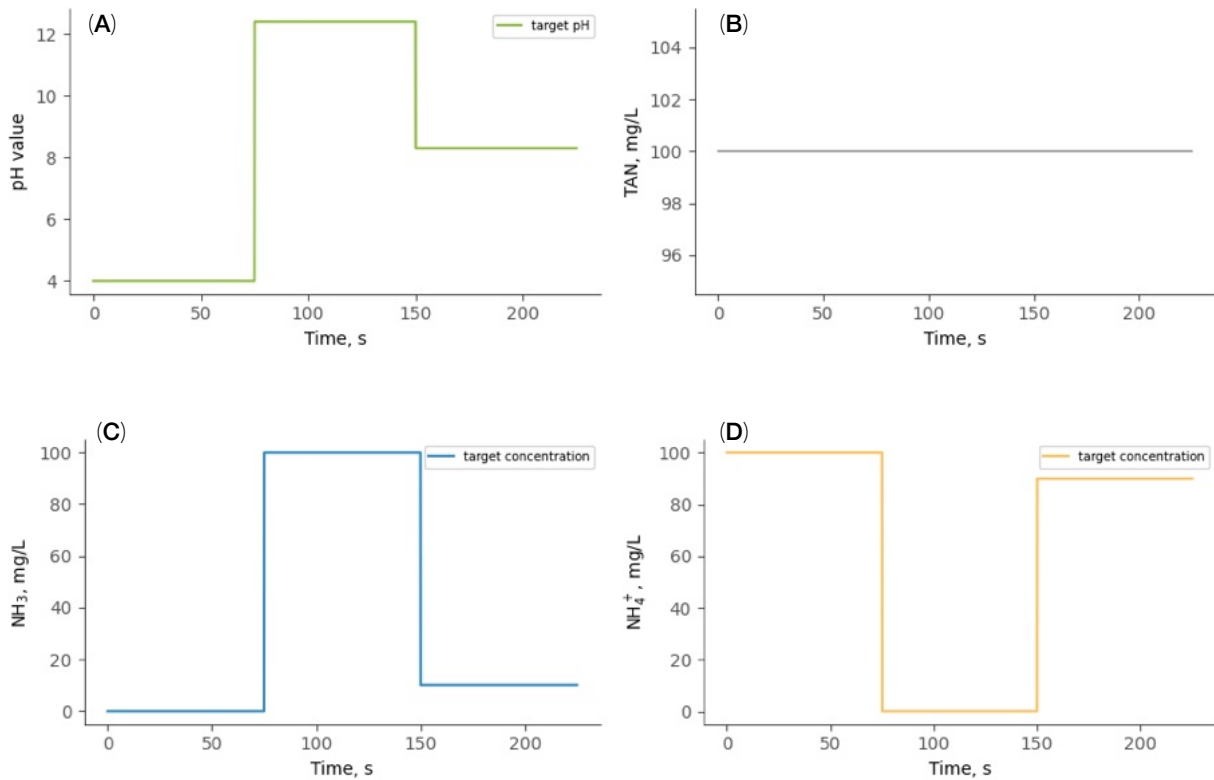
**Equations (1) and (2)** describe the starting point for our simulation. For simulating the TAN equilibrium, the base and acid are  $\text{NH}_3$  and  $\text{NH}_4^+$ . To determine the total parameter (TAN), the pH value as well as one of the analyte species is monitored. Hence, transforming **equation (2)** and combining it with **equation (1)** results in the general description of the base or acid species:

$$[\text{Base}] = \frac{[\text{Total}]}{1 + 10^{\text{pH} - \text{pK}_a}} \quad \text{and} \quad [\text{Acid}] = \frac{[\text{Total}]}{1 + 10^{\text{pK}_a - \text{pH}}} \quad (3) \text{ and } (4)$$

And more specifically for the TAN simulation experiment:

$$[\text{NH}_3] = \frac{[\text{TAN}]}{1 + 10^{\text{pH} - \text{pK}_a}} \quad \text{and} \quad [\text{NH}_4^+] = \frac{[\text{TAN}]}{1 + 10^{\text{pK}_a - \text{pH}}}$$

For the simulation (cf. **Figure 1**), we assume a target TAN concentration of 100 mg/L and that the pH value and ammonia concentration are monitored over time. The ammonium concentration  $\text{NH}_4^+$  and subsequently the TAN concentration will be calculated based on **equations (4) and (2)**. Then, we can compare how much the error propagation affects the calculated total parameter concentration when calculated from two individual sensors with different sensor response times.



**FIGURE 1** User-defined pH values (A) and total parameter concentration (B) over time and resulting target concentration for the individual ammonia species (C-D). For the simulation of the target concentrations. An infinitely fast sensor response is assumed, resulting in the respective step function.

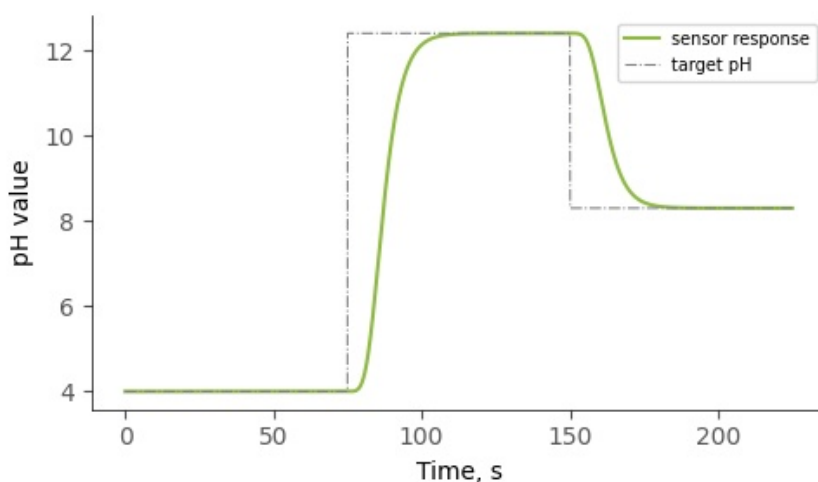
However, sensors do not respond infinitely fast. In reality, sensors have a certain response time, usually defined as  $t_{90}$  time. To account for this delayed sensor response, we simulated a sigmoidal response pattern, which can be described by a **Gompertzian curve** for sigmoidal time series:

$$f(t) = a \cdot \exp(-\exp(b - c \cdot t)) + d \quad (5)$$

## PARAMETERS

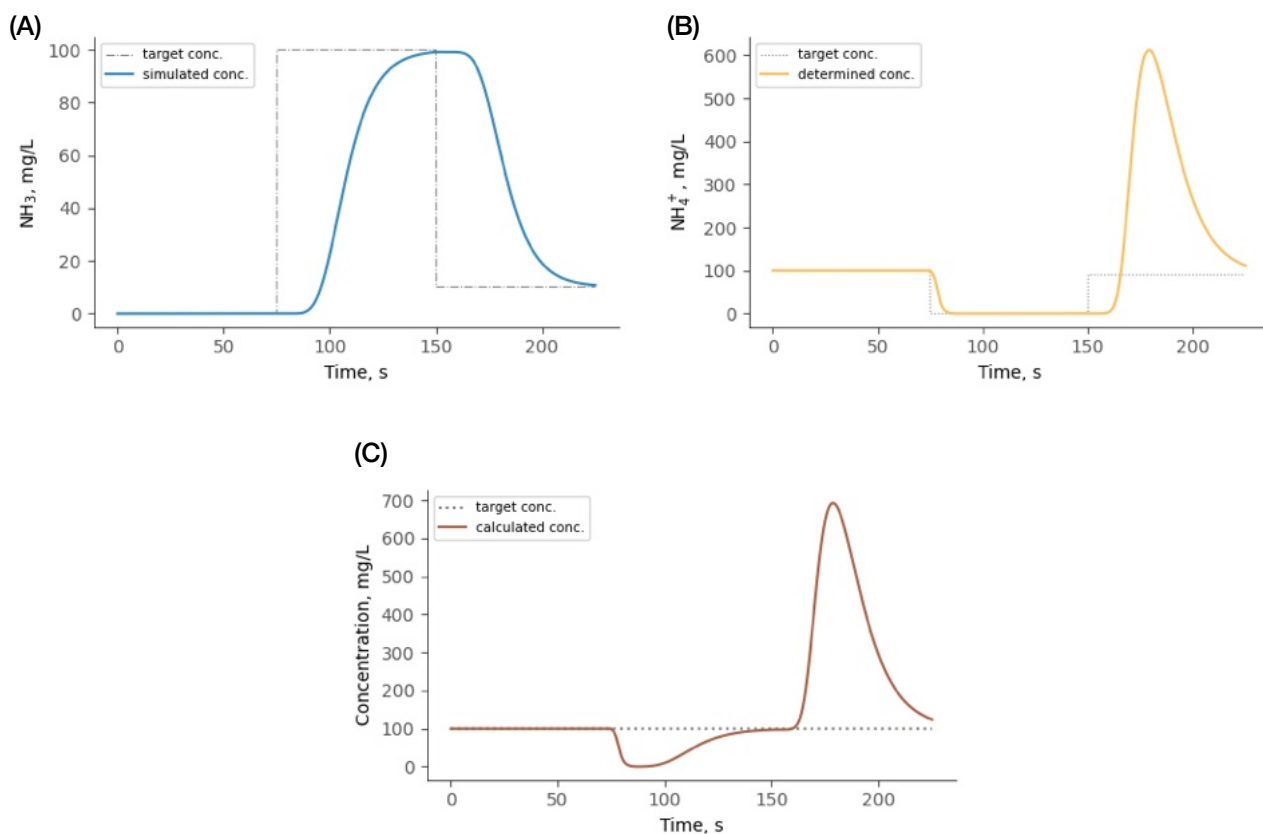
- $a$  is the difference between the actual and the target concentration or pH value.
- $b$  sets the displacement along the x-axis (translates the graph to the left or right). The sensor resolution feeds into this parameter with  $b = \log(\log(1/\tau))$  and  $\tau$  being the sensor resolution in seconds.
- $c$  specifies the growth rate or y-scaling of the sigmoidal curve. The sensor resolution and response time both feed into this parameter with  $1/t_{90} \cdot [b - \log(\log(1/(1-r)))]$ , where  $t_{90}$  is the sensor response time,  $r$  is the sensor resolution, and  $b$  is the displacement factor along the x-axis as previously defined.
- $d$  specifies the displacement along the y-axis. For our simulation, it sets the initial plateau from which the sigmoidal curve is simulated.

Applying the Gompertz curve to the target pH fluctuation, we can simulate the delayed response of the pH sensor depending on its specifications as shown in **Figure 2**.



**FIGURE 2** Sensor response of an ideal and real pH sensor. The response of the target/ideal sensor response with an infinitely fast response time is shown as a grey dash-dotted line, while the response of the real sensor with a  $t_{90}$  time of 20 s is shown as a green solid curve.

Based on the target pH value and total ammonia concentration, a sensor response for the measuring sensor including its sensor response is simulated (cf. **Figure 3**). In this example, the measuring sensor is an  $\text{NH}_3$  sensor. The  $\text{NH}_4^+$  concentration is then calculated at each time point based on the sensor response of the  $\text{NH}_3$  sensor and the simulated reading of the pH sensor. The TAN concentration is then calculated as the sum of the  $\text{NH}_3$  and  $\text{NH}_4^+$  at each time point. The following plots show the resulting error propagation. It has to be stated that the same procedure applies when  $\text{NH}_4^+$  is selected as the measuring sensor and  $\text{NH}_3$  is calculated.



**FIGURE 3** Simulation of error propagation for the calculation of TAN. The pH (same as Fig. 2) and NH<sub>3</sub> (A) are monitored over time considering the respective sensor response times. The  $t_{90}$  response times of the real sensor pH and NH<sub>3</sub> are 20 s and 60 s, respectively. From these simulated values the NH<sub>4</sub><sup>+</sup> concentration (B) and further TAN (C) can be calculated.

It should be mentioned that we also considered the **Nernst equation** to convert the sensor signal to the analyte concentration and vice versa.

$$E = E_0 - 2.303 \cdot R \cdot \frac{T + 273.15 \text{ K}}{n \cdot F} \cdot \text{pH} \quad (6)$$

## PARAMETERS

- $E_0$  is the standard reduction potential in V
- $R$  is the universal gas constant 8.314 in J/mol·K
- $F$  is the Faraday constant or the charge of an electron 96485 in C/mol
- $n$  is the number of electrons involved in the redox reaction
- $T$  is the temperature in °C
- **pH** is the apparent pH value

## Preparation of function inputs

All parameters required for the simulation are listed and functionally described below. Global parameters are fixed and cannot be changed in the software, while others are subject to analysis and can therefore be varied in the software interface.

### Overview of global (fixed) parameters

**TABLE 2** Auxiliary parameters used for simulation.

Parameter	Value	Description
tsteps	$10^{-3}$	Time steps for simulating the time series; in seconds
R	8.314	Universal gas constant; in J/mol*K
F	96485	Faraday constant; in C/mol
n	1	Number of electrons involved in the redox reaction
ph_res	$10^{-3}$	Resolution of the pH sensor
ph_E0	0.22	Standard potential of the reference electrode of the pH meter; in V
sens_res	$10^{-9}$	Resolution of the measuring sensor
conc_calib	1	Reaction quotient for calibrating the measuring sensor (here $\text{NH}_3$ )
sens_E	0.09	Potential of the measuring sensor at a certain calibration concentration ( <i>conc_calib</i> ); in V

### Overview of the sensor parameters used in the TAN example

**TABLE 3** Sensor specifications used for the simulation. Can be adjusted by the user on the initial page when starting the software.

Parameter	Value	Description
Plateau time	75.0	Plateau time as the time a pH value is supposed to be stable; in seconds
pH value(s)	4, 12.4, 8.3	List of pH values given as float number, separated by a comma “,”
Response time	20, 50	Response time of the pH and measuring sensor; values are given as float numbers in seconds
c(TAN)	100	Total parameter concentration (here TAN) given as a float number in mg/L. It is possible to add a list of target concentrations separated by a comma “,”
pK <sub>a</sub>	9.25	Acid dissociation constant as the equilibrium constant of the system to be studied. For TAN we assumed a pK <sub>a</sub> of 9.25, while for TDS we suggest using a pK <sub>a</sub> of 7.



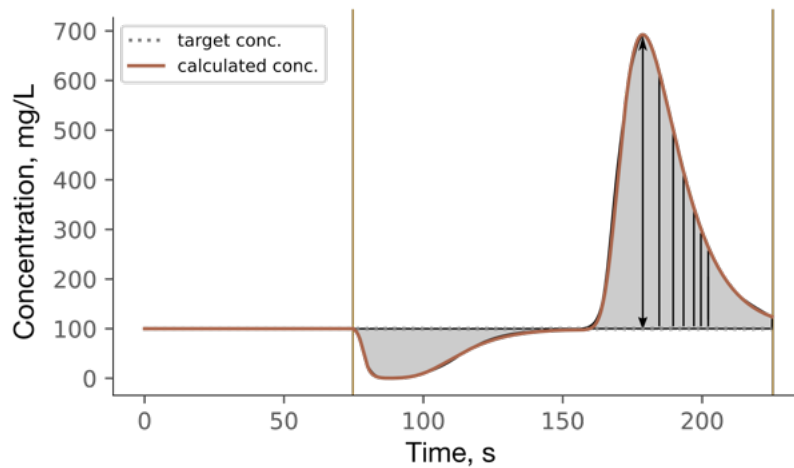
## Results of functions (Simulation Output)

Simulation of the error propagation for the calculated analyte and the total parameter displayed in the response signal over time (and changing pH value).

- Response curves can be exported and saved as images (either directly by right-mouse click and selecting export or via the button “SAVE ALL”).
- Individual (measured and simulated) sensor responses can be exported as text files via “SAVE REPORT”.
- Error expressed as overall deviation from the target concentration of the total parameter can be calculated from the total parameter plot.

### Calculating the overall error

An error is defined as the deviation between the measured and the target value. While one can calculate the difference between the two values at a certain time point, we were more interested in the overall error over time (cf. **Figure 4**). In our simulation, this equals the area between the sensor response and the target concentration of the total parameter.



**FIGURE 4** Simulated sensor response of the total parameter (TAN) as a result of the error propagation of the measured and calculated (pH dependent) analyte. The deviation of the calculated concentration (solid brown line) to the target concentration (black, dotted line) at a certain time is indicated as a (vertical) black line. Hence, the total error within a given time range is given as the area underneath the response curve. The total error is marked as a grey area.

To determine the error of the simulated measurement, we determine the integral of the sensor response compared to the integral of the target concentration within a user-defined time range. The integral of the total signal resulting from the individual sensor responses is determined using Simpson’s rule, a numerical integration of discrete but equally spaced sample points. Simpson’s rule is based upon a quadratic interpolation as follows:

$$\begin{aligned}\int_a^b f(x)dx &\simeq \frac{1}{3} \cdot h[f(a) + 4 \cdot f(\frac{a+b}{2}) + f(b)] = \\ &= \frac{b-a}{6} [f(a) + 4 \cdot f(\frac{a+b}{2}) + f(b)]\end{aligned}\tag{7}$$

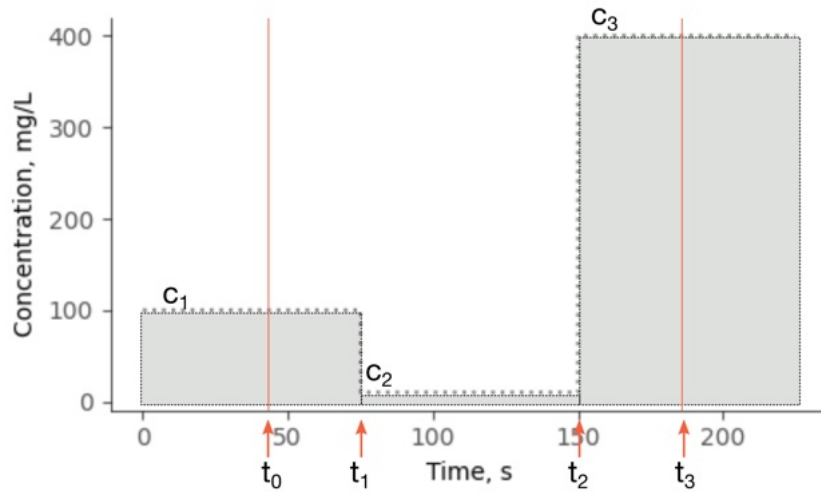
where  $h = (b - a)/2$  is the step size.

To determine the integral of the target concentration, the area of the individual steps is determined as the sum of individual rectangles:

$$\int_a^b f(x)dx \simeq c_1(SUM) \cdot (t_1 - t_0) + c_2(SUM) \cdot (t_2 - t_1) + \dots \quad (8)$$

where  $c_i(SUM)$ , with  $i = 1, 2, \dots$ , is the user-defined target concentration of the total parameter and  $t_j$  with  $j = 1, 2, \dots$ , are the individual integration limits sub-dividing the target concentration within the user-defined integration range into rectangles (cf. **Figure 5**).

The difference between the area of the simulated total concentration and the area of the target concentration equals the overall error.



**FIGURE 5** Explanatory plot to illustrate the integration of the target concentration of the total parameter concentration. Since we assume infinitely fast responding sensors for the target concentrations, we also obtain a step function for the total parameter concentration. When the user selects integration limits at times  $t_0$  and  $t_3$ , the total integral can be calculated as the sum of the area of each rectangle.

## References

- (1) PyroScience GmbH | pH meters. [pyroscience.com/en/products/ph](https://pyroscience.com/en/products/ph) (accessed 2023-06-12).
- (2) Presens | Optical pH Sensors. <https://www.presens.de/products/ph/sensors> (accessed 2023-06-12).
- (3) Krohne GmbH | pH sensors and measuring systems. <https://krohne.com/en/products/process-analytics/analytical-sensors-measuring-systems/ph-sensors-measuring-systems> (accessed 2023-06-12).
- (4) Unisense | pH microelectrode for research applications. Unisense. <https://unisense.com/products/ph-microelectrode/> (accessed 2023-06-12).
- (5) Metrohm | 780/781 pH/ionmeter. [https://www.metrohm.com/da\\_dk/products/ph-ion-measurement/Laboratory-ion-pH-meters.html](https://www.metrohm.com/da_dk/products/ph-ion-measurement/Laboratory-ion-pH-meters.html) (accessed 2023-06-12).
- (6) Hach | A-ISE sc Low cost ISE Ammonium probe (immersion) with RFID, 10 m cable. <https://ie.hach.com/a-ise-sc-low-cost-ise-ammonium-probe-immersion-with-rfid-10-m-cable/product?id=26371051351> (accessed 2023-06-12).
- (7) Thermo Scientific Orion | High-Performance Ammonia Electrode - pH and Electrochemistry, Probes and Electrodes. <https://www.fishersci.com/shop/products/orion-high-performance-ammonia-electrode/13643500> (accessed 2023-06-12).
- (8) Metrohm | NH<sub>3</sub>-selective gas membrane electrode (low conc.). <https://www.metrohm.com/en/products/6/0506/60506100.html> (accessed 2023-06-12).
- (9) Unisense | H<sub>2</sub>S Microsensor for hydrogen sulfide research. Unisense. <https://unisense.com/products/h2s-microsensor/> (accessed 2023-06-12).
- (10) SulfiLogger | H<sub>2</sub>S sensor. <https://sulfilogger.com/sulfilogger-sensor/> (accessed 2023-06-12).

# Operating Instructions (step-by-step-guide)

## Where to download the software application and how to get started?

To run the software application, the source code must be downloaded from GitHub. It is publicly available via the following link: <https://github.com/silviaelisabeth/SensorResponse-inSilico>.

Download the application folder including the subfolder (**PICTURE**). To run the software application, open the file named [SensinSilico.py](#) with a Python editor or IDE of your choice and run the script. We suggest using PyCharm, VS Code or the Terminal/Command Prompt as we tested the software application with PyCharm and VS Code 1.74.2 (Universal) for software development and testing.

For the software application to run properly, leave all the files in the folder together without restructuring or removing any parts.

## Requirements

When using the source code, python 3 and the following Python packages are required for execution:

*numpy (version 1.24.2), pandas (version 1.5.3), PyQt5 (version 5.15.9), pyqtgraph (version 0.13.3), scipy (version 1.10.1), and seaborn (version 0.12.2).*

The package versions listed are the versions used when the software is released. Older versions might cause errors and bugs, whereas newer versions should not make much difference. The software has been tested for both MacOS (MacBook Pro – Apple M1, Version 13.4) and Windows (Version 10.0.19045). If you have difficulties running the software, please do not hesitate to reach out (cf. **Points of Contact**).

## IntroPage | Define Parameter

Upon executing the script [SensinSilico.py](#) a separate window should open displaying the IntroPage of the software application (cf. **Figure 6**).

In the upper part of the IntroPage the global parameters for the simulation must be defined. Furthermore, the properties of the two single sensors have to be defined, where sensor 1 always defines the pH sensor. For the second sensor, either a sensor can be defined as part of the  $\text{H}_2\text{S}/\text{HS}^-$  or  $\text{NH}_3/\text{NH}_4^+$  equilibrium or a parameter of another weak monovalent acid/base equilibrium can be selected. In the latter case, the axes of the simulation plots are generically labeled acid/base, since only TAN and TDS were included as standard examples. A list of target pH values to be studied must be defined for the pH sensor. As response time, assume a common response time for commercially available pH sensors as presented in **Table 1**.

For sensor 2, specify which sensor should be used for theoretical measurement. For example, if you want to study the TAN equilibrium, select either  $\text{NH}_3$  or  $\text{NH}_4^+$  as your measurement sensor. Once you have specified the equilibrium, you need to add the characteristics, such as the expected sensor response time and the  $\text{pK}_a$  of your system. We suggest a  $\text{pK}_a$  of 9.25 for TAN, and 7.0 for TDS respectively. To finish the preparation, add a concentration for the sum parameter (in mg/L) or even a list of target concentrations, separated by a comma.

IntroPage | define parameter for initial plot

(1) Sensor specifications

Parameter Settings for Simulation

Plateau time  s

Concentration(s) sum parameter  mg/L

Sensor - 1

Name

pH value(s)

Response time  s

Sensor - 2

Name

Response time  s

pKa

(2) Directories for loading and saving

Define Directories

Load meas. file

Storage path

(3) Continue to Simulation

< Back Next > Cancel

**FIGURE 6** Front page of SensinSilico. Here, the initial sensor specifics (1) such as the response time of the pH sensor and the monitored analyte must be defined. Furthermore, specify the species that shall be monitored and set the  $pK_a$  value of the acid/base equilibrium. Optional, one can load previous simulations or already specify the directory where to save the results (2). When all required parameters are defined, press NEXT (3) to continue to the simulation page.

In case any previously simulated experiment shall be imported or any output shall be saved, define the directory using the respective buttons in the lower part ("LOAD MEAS. FILE" and "STORAGE PATH"). Pressing "NEXT" will lead you to the second page of the software wizard. In case it is not possible to access the second page, make sure that all required data is added/defined.

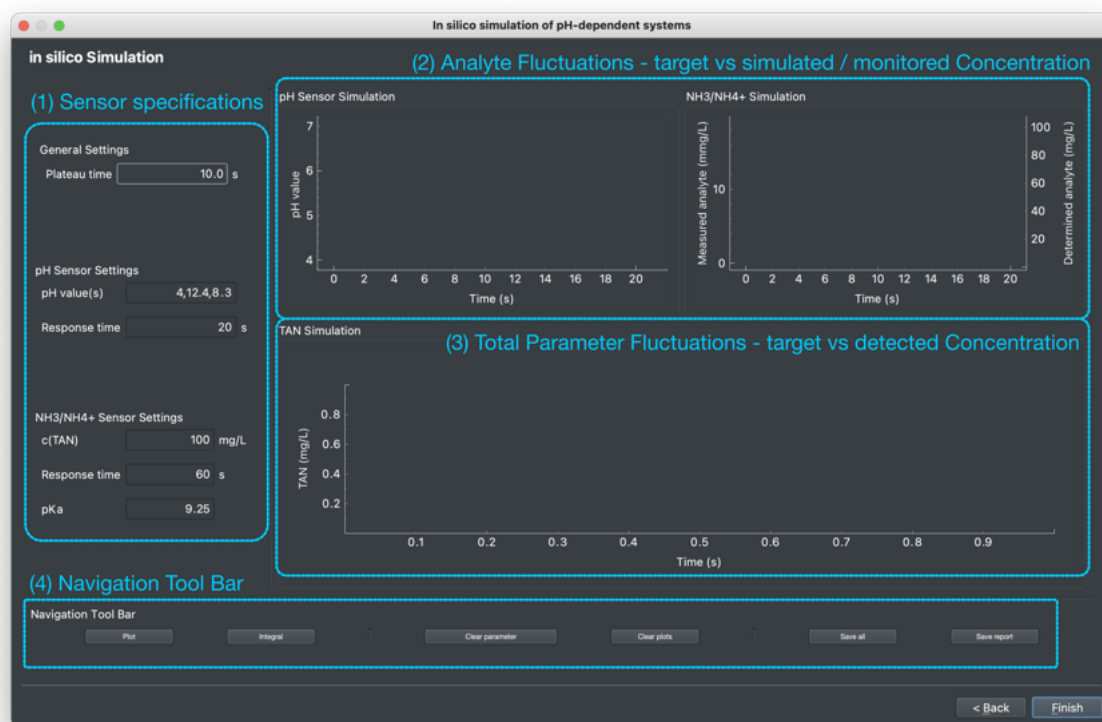
## REQUIRED INPUT

<b>Plateau time</b>	Time period of how long one pH value or sum parameter concentration is assumed to be stable; time given as float number in seconds
<b>Concentration(s) of sum parameter</b>	Target concentration(s) of the total parameter to be simulated. Numbers given as float numbers separated by a comma
<b>Sensor 1</b>	It is assumed that the first sensor is always a pH sensor
▶ pH value(s)	Target pH values (between 1-14); values given as float numbers separated by a comma
▶ Response time	Response time of a pH sensor; here expressed as $t_{90}$ in seconds
<b>Sensor 2</b>	Measuring sensor which usually has a different response time than the pH sensor
▶ Name	Choose either $NH_3$ or $NH_4^+$ for a TAN simulation or $H_2S$ or $HS^-$ for a TDS simulation
▶ Response time	Response time of the measuring sensor expressed as $t_{90}$ in seconds
▶ $pK_a$	Acid dissociation constant describing the equilibrium constant of the system to be studied

In the software, use points (".") for decimal separators.

## Simulation Page

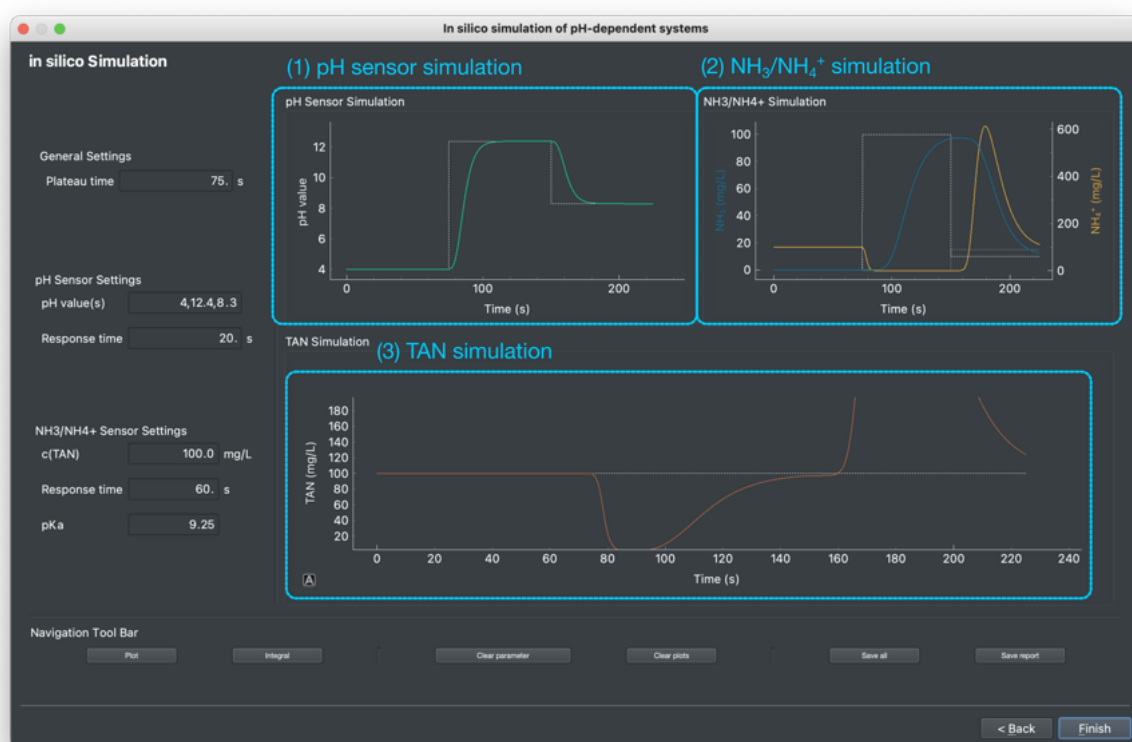
### Generate the simulation



**FIGURE 7** Overview of the Simulation Page. On the left side (1) the sensor specifications are listed as previously defined. They can be adjusted in case a comparative analysis should be simulated. On the top right (2), the individual analyte concentrations over time are displayed. The left panel shows the target and simulated pH fluctuation while the right panel will display the measuring sensor and its corresponding calculated acid/base form. The latter is a pure simulation based on the calculations described in the Background chapter. The panel at the bottom (3) will show the target and simulated fluctuations of the total parameter concentration. At the bottom (4), a navigation toolbar will support the basic operational steps in the software.

In the left panel, all previously defined parameters are summarised and the pH-dependent equilibrium that you intend to study is specified. On the right, three panels display the individual sensors (two plots at the top) and the resulting total parameter (bottom plot). In the bottom panel, the navigation toolbar presents an overview of what you can do with your simulation results.

If all required parameters are added/defined, press “PLOT” in the Navigation toolbar to start the simulation. The simulation for the individual sensors will be displayed in the top two plots and the resulting total parameter will be displayed in the bottom plot (cf. **Figure 7**).



**FIGURE 8** Example of a simulation for total ammonia nitrogen (TAN). The top left panel (1) displays the pH values defined by the user (here 4, 12.4 and 8.3) and the individual species for the defined acid/base equilibrium (2). The bottom panel (3) then shows the simulated total ammonia nitrogen fluctuations compared to its target concentration. The target concentrations are displayed in white dotted lines while the simulated sensor responses are displayed as coloured solid curves.

Starting with a pH of 4, there is a pH change to 12.4 and subsequently to 8.3 with a plateau time of 75 seconds (**Figure 8(1)** – pH Sensor Simulation). As a white dash-dotted line, a step function indicates the target pH change assuming instantaneous pH change. The simulated real sensor response (considering the defined sensor properties) is shown as a solid green line.

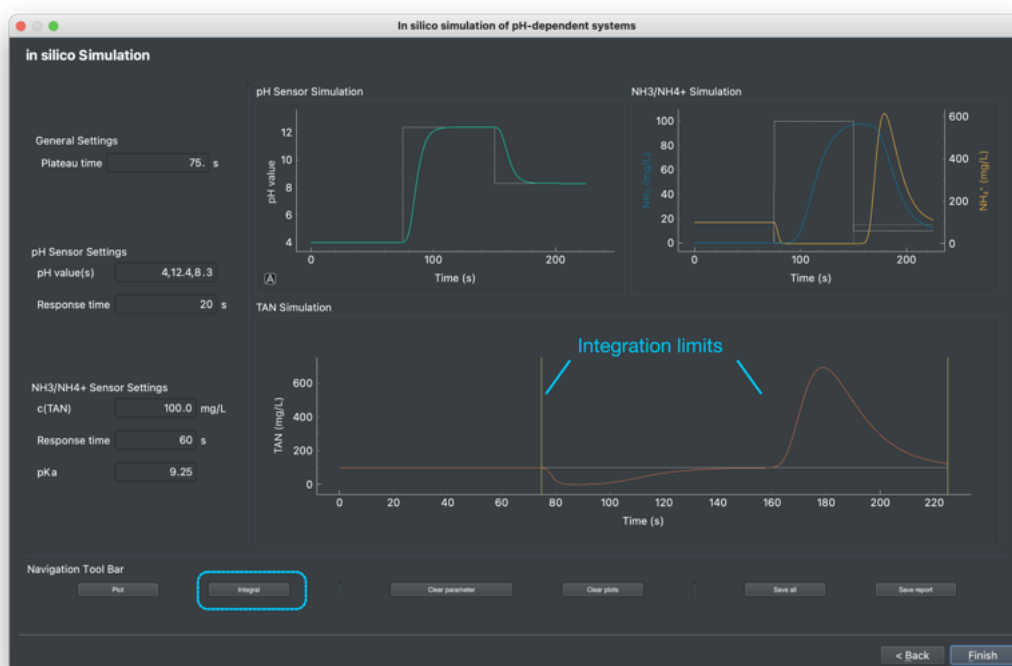
In the panel to the right (cf. **Figure 8(2)**;  $\text{NH}_3/\text{NH}_4^+$  Simulation), both analytes are displayed together using individual y-axes. On the primary y-axis, the analyte is displayed that has been specified by the user as the measuring sensor (here  $\text{NH}_3$ ; solid blue line), while the corresponding calculated analyte (here  $\text{NH}_4^+$ ) is displayed as an orange solid line using the secondary y-axis. The white step functions represent the target function assuming an instantaneous change according to the pH change while the coloured lines consider a delayed sensor response.

In the panel below (cf. **Figure 8(3)**; TAN Simulation), the total analyte concentration over time (and changing pH) is displayed. While the white dash-dotted line represents the target concentration as defined by the user, the brown line displays the total analyte concentration when summing up the two individual sensor concentrations visualising the error propagation due to delayed sensor response. The initial view is set to focus on the target concentration  $\pm 90\%$ . To display the entire sensor response, the user may specify the scene by right-clicking in the plot or by pressing the [A] in the bottom left corner. To zoom in, first select the 1-button Mouse Mode (right-click option). Afterwards, you can zoom in by drawing a yellow rectangle across the area of interest with your cursor.

## Determining the overall error of the total parameter

To determine the overall error of the total parameter caused by the different sensor response times, the area between the simulated total response curve and the target concentration must be determined. Further details on the theory and calculation steps can be found in the theory section (**Background | Application's Functions**).

To define the time range in which the overall error shall be calculated, simply click on the plot while holding the COMMAND/WINDOWS key. A vertical line appears in the plot defining the time range for integration. The user may fine-tune the vertical line which can easily be dragged along the x-axis. Once the time range is selected, press the "INTEGRAL" button in the Navigation toolbar (cf. **Figure 9**). A separate window will pop up, summarising the Integration range, the target pH and total concentration together with the integral and the resulting error (cf. **Figure 10**).

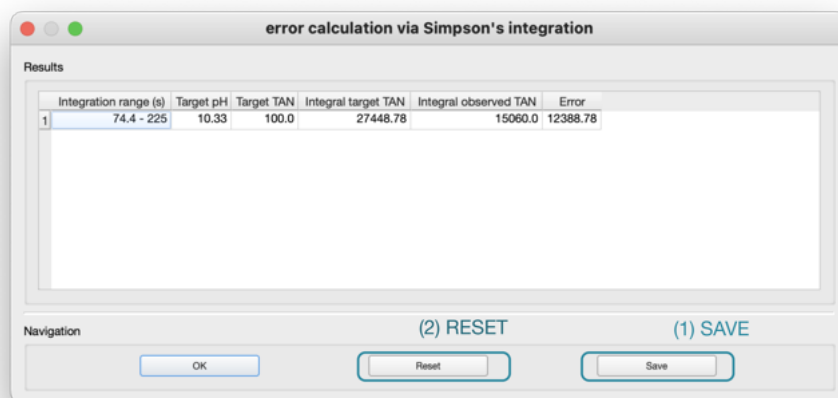


**FIGURE 9** For calculating the overall deviation between the target and simulated total parameter concentration, one needs to specify the time range within which the deviation shall be calculated. This can be done by clicking into the plot while pressing the command/windows key. The vertical lines can then be dragged along the x-axis to fine-tune the integration range. When finished, press integral to calculate the overall error.

## Save output

It is possible to save individual results such as only the integral table or the simulation graph as a text file or the entire simulation including text files and plots.





**FIGURE 10** Pop-up window upon pressing integral. Here, a summary of the target ph and the total parameter is shown, as well as the calculated integrals of the target vs. the simulated total parameter and its resulting error. The buttons (1) and (2) allow the separate saving of the integral and resetting of the entire error calculation.

If only the integral table is to be saved, press the “SAVE” button in the separate pop-up window (cf. **Figure 10(1)**) and define the directory and file name under which the simulation is to be saved.

To save only the simulation graph, select “EXPORT” after right-clicking in the respective simulation plot and define the directory and file name under which the simulation is to be saved. Note that this graph will be saved as it is, i.e., with a dark background. If only the calculations are to be saved, but no images, select “SAVE REPORT” (cf. **Figure 11(2)**). Then the entire simulation as well as an extra file summarising the previously defined integration and error calculation will be saved. With “SAVE ALL” (cf. **Figure 11(1)**), both the text files and the diagrams will be saved in the specified directory.



**FIGURE 11** Additional operations in SensinSilico to either reset the plots (4) and parameters (3) for the current simulation or to save the output (1-2). For the latter, the user can save only the simulation and potential error calculations as a text file (2) or save the simulation plots (1).

## Terminate and Restart Operations

### *Reset function*

The parameters and plots must be cleared individually by pressing “CLEAR PARAMETER” and/or “CLEAR PLOTS” in the Navigation toolbar in the main window (cf. **Figure 11(3-4)**). In case the simulation should remain but only the integration table should be deleted, press “RESET” in the respective pop-up window (cf. **Figure 10(2)**).

### *Terminate application*

To close the application, either press “FINISH” in the Navigation toolbar or close the main window with the short-cut **COMMAND + Q** when working on a Mac or **ALT + F4** for Windows.