Sequencer

(An Introduction)

ZAHNER SCIENTIFIC INSTRUMENTS

06/2022

1 Introduction

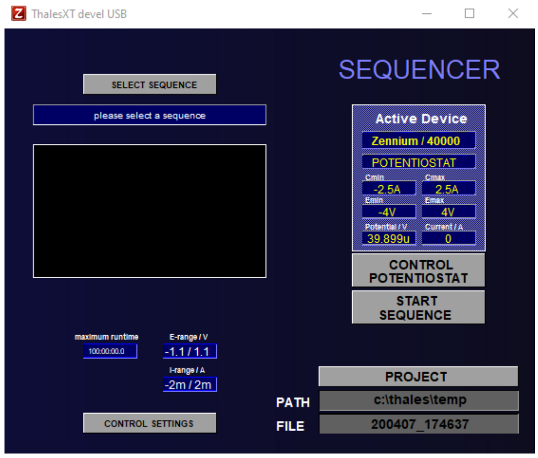
In research labs and industries, simple electrochemical processes are repeated multiple times to test different systems under investigation. The repetition of the following DC processes can be easily automated using the “Sequencer” software.

- Potential ramp
- Current ramp
- Constant potential/current vs time
- Hold at OCP

2 Starting sequencer program

Click on the *EXE* icon in the classic mode of the *Thales* software and open the *sequencer.rtm* file. This will open the “Sequencer” window. The .rtm file is available in folder *c:\Thales*.

Thales software – Classic mode



Sequencer

In the “Sequencer” window, sequences are uploaded for the measurement. Different sample sequences are provided in folder *c.\thales\script\sequencer\sequences*. A sequence can be uploaded in the sequencer window via SELECT SEQUENCE. The user can also write a new sequence and upload it. The appropriate potential and current ranges can be modified via CONTROL SETTINGS. After loading the sequence and setting the appropriate potential and current ranges, start the sequence by clicking on the START SEQUENCE.

The PROJECT indicates the path where the measured data file will be saved. A default filename is set to DATE_TIME format. The “Sequencer” actively save the measured data (in blocks) during the measurement and does not wait for the measurement to end to generate output files. A sequence can be run for a maximum time of 1000 hours.

3 Writing a sequence

A sequence is written in the THALES EDITOR window and follows a basic structure (provided below)

Sequence format

```
start_cycle    'start sequence
samples(20)    'measure 20 reading per second
do_loop(5)     'do 5 loops
.
.              'sequence body
.
end_loop       'end loop
end_cycle      'end sequence
```

Commands are written in **pink** and comments are written in **green** color. These comments are not processed by the sequencer. More information about color coding is provided in the manuals **SCRIPT** and **Script - an introduction**. The limit for the sample frequency is 1 mHz to 100 Hz and in sequences, the samples (per second) are provided in the format of 1,2,5…. so on.

3.1 Potential ramp

Potential ramp can be applied by providing the *starting* and *end potential* value in *Volts* (*V*). The *slope* of the ramp or *time* (for measurement) can be provided in *V/s* or *s*, respectively.

ramp_pot_s(*starting potential,end potential,slope,Imin,Imax*) `'ramp with defined slope`

OR

ramp_pot_t(*starting potential,end potential,time,Imin,Imax*) `'ramp with defined time`

Imin (A) and *Imax* (A) are optional and are used to provide the lower and upper limits of the current. The current range is provided to avoid the unwanted high currents which may damage the system under investigation. If such a range is not necessary then it can be omitted from the code.

3.1.1 Potential ramp starting from actual potential:

A potential ramp can also be initiated from the actual potential of the potentiostat to the defined end potential. This is possible with the following code

ramp_pot_t(*PACT,end potential,time*)

Here, *PACT* represents the actual potential of the sample/system under investigation (i.e., OCP when no potential is applied). This code can also be modified for the time and slope variable. In addition, current can be limited during the ramp.

A potential ramp, from 0 V to 100 mV with a period of 1 s will have a line code as given below.

```
ramp_pot_s(0,0.1,0.1)     'do ramp in 0.1V/s (100mV/s) – without current range -

ramp_pot_t(0,100m,1,-10m,100m)     'do ramp in 1 second (with current range)
```

Here in the time-defined ramp, an optional current range of -10 mA to 100mA is also provided.

3.2 Current ramp

The current ramp can be applied by providing the *starting* and *end current* values in *Volts* (*V*). The *slope* of the ramp or *time* (for measurement) can be provided in A/*s* or *s*, respectively.

ramp_cur_s(*starting current,end current,slope,Emin,Emax*)

OR

ramp_cur_t(*starting current,end current,time,Emin,Emax*)

Here, *Emin* and *Emax* values are optional and provide the potential range for the current ramp.

3.3 Hold at potential/current

A constant potential/current can also be applied for a defined period. The line code of this constant potential or current is provided below

```
hold_pot(potential,time)

hold_pot(PACT,time)

hold_cur(current,time)

hold_cur(CACT,time)
```

CACT is the actual current flowing through the system under investigation. Here, *Imin* and *Imax* can be incorporated for the potential sweep and *Emin* and *Emax* for the current sweep.

3.4 Hold at OCP

The system under investigation can also be held at the open circuit potential for a defined period. For this use the code provided below

ocp(time)

4 Sample sequences

In this section, different sample sequences are provided which can be used in the Sequencer program.

Sequence 1

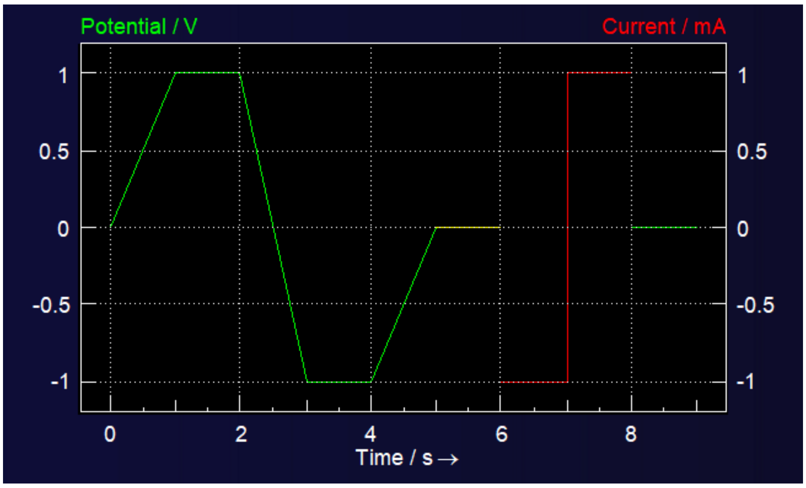
A simple sequence, where the potential ramps from 0 V to 1 V in 10 seconds and then this ramp is repeated 50 times, is provided below.

```
start_cycle            'start sequence
samples(20)            'measure 20 reading per second
do_loop(50)            'do 50 loops
ramp_pot_t(0,1,10)     'starting potential=0, end potential=1, time=10 s
end_loop               'end loop
end_cycle              'end sequence
```

Sequence 2

The sequence is visualized in the graph on the right side. Here, the controlled potential ramps are shown with green color, holding at OCP with yellow, and the current ramp with red color. Starting from OCP measurements the sample measured per second is set to 10.

```
 start_cycle
   samples(20)
   do_loop(50)
      ramp_pot_t(0,1,1)
      hold_pot(1,1)
      ramp_pot_s(1,-1,2)
      hold_pot(-1,1)
      ramp_pot_s(-1,0,1)
         samples(10)
         ocp(1)
       hold_cur(-0.001,1)
       hold_cur(1m,1)
       hold_pot(PACT,1)
   end_loop
end_cycle
```



In a sequence, between two commands a “dead time” of around 100ms is present. During these circa 100ms the sequencer will not record any data. Hence the user must use a recommended minimum of 1 second time for each command. For example, a very fast pulsing in the sequencer software with a pulsing time of less than 1s will not be properly carried out.

Sequence 3

In the sequence, the user can use variables despite the intended values. These variables can then be defined between *start_variables* and *end_variables*. This way user does not have to change the values through the complete sequence and only changing the values for the *variables* will be sufficient.

```
start_cycle
  samples(20)
  do_loop(5)
    ramp_pot_t(POT0,POT1,TIM1)
    hold_pot(POT1,TIM2)
    ramp_pot_t(POT1,POT0,TIM1)
  end_loop
  ramp_cur_t(CUR1,CUR2,TIM1)
  ramp_cur_t(CUR2,CUR1,TIM1)
end_cycle


start_variables 'DEFINE VARIABLE
  POT0=0
  POT1=1
  TIM1=1
  TIM2=5
  CUR1=0
  CUR2=10m
end_variables
```

Restrictions: Currently only 3 variables TIM, POT, and CUR can be used for time, potential, and current respectively. The difference between similar variables (i.e., potential variables) can be made by using a number with the variable. Numbers from 0-99 are allowed in the Sequencer.

The comments are not allowed in sequences. The Sequencer will not load the sequences where the comments are also written in the sequences. In this manual, comments are written in sequences to explain the function of coding

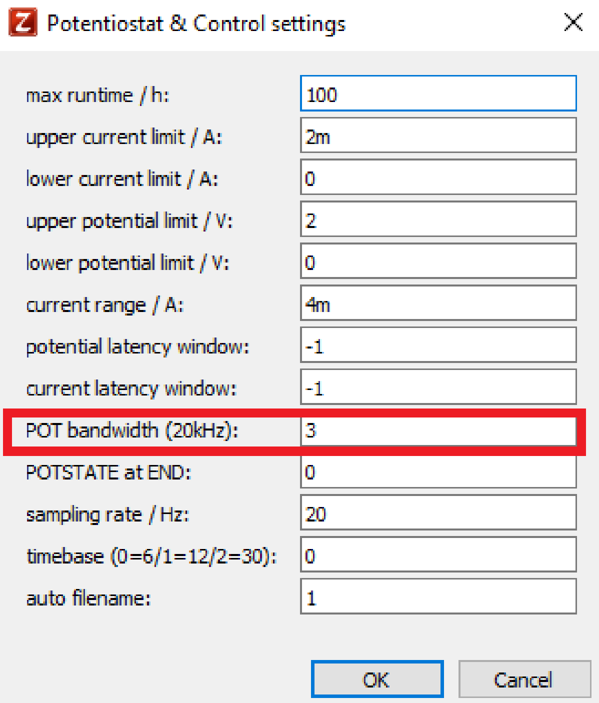
5 Global current and potential limits

In the sequence code outside the loop, different “control settings” can also be coded. This will automatically change the control settings when a sequence is uploaded and the user does not have to modify the potentiostat setting.

```
start_cycle    'start sequence
samples(20)    'measure 20 reading per second
do_loop(5)     'do 5 loops
.              'sequence body
.
end_loop       'end loop
end_cycle      'end sequence
start_online
cur_hi=2m      'upper current limit / A - (2 mA)
cur_lo=0       'lower current limit / A - (0 mA)
pot_hi=400m    'Upper potential limit / V - (2 V)
pot_lo=100m    'lower potential limit / V - (0 V)
cur_ra=4m      'current range / A – (4mA)
pot_of=-1      'potential latency window (POT-OFF)
               'pot_of=-1 → do not turn off
               'pot_of= t → POT-OFF after “t” seconds, i.e.,
               'pot_of=0, turn off immediately
               'pot_of=2, turn off after 2 seconds if the potential does not
               decrease to the desired potential range in the 2 seconds
cur_of=-1      'current latency window (Current off) –Same as POT-OFF
end_online
```

“pot-of” and “*cur-of”* decide what to do if the potential and current values exceed the described limits respectively.

During current ramps, if the potentiostat show noise during measurement then in the CONTROL SETTINGS, change the POT bandwidth to 3.



6 Incorporating a sequence in SCRIPT

The “Sequencer” only runs the DC tests. If the user wants to carry out different DC and AC measurements in a single run then the sequencer can be used with Zahner’s script program. For this, the user can write the sequence(s) and then save the sequences in the folder *c:\thales\script\sequencer\sequences*. Once the sequence(s) are saved then the user can call them in the script program and then run the sequences before or after AC measurements (i.e., EIS) or other standard measurements (CV).

Example:

The script provided on the next page contains the following scheme

1. Start
2. Sequence 1
3. EIS measurement
4. Sequence 2
5. Script body
6. End

```
SCRIPT1
SEQUENCE%=1                'load c:\thales\script\sequencer\sequences\sequence01
 gosubLOADSEQUENCE         'go to sequencer module
 gosubEXECUTESEQ           'do sequence
 if(SEQERROR%=0)trueif
 endif
MEAS_OPEN_EIS(65,"c:\thales\script","eis")      'open EIS rule file
MEAS_EIS                                        'record EIS spectrum
MEAS_SAVE_EIS(65,"@c:\thales\script”,"eis0")    'save measured EIS
     SEQUENCE%=2           'load c:\thales\script\sequencer\sequences\sequence02
     gosubLOADSEQUENCE     'go to sequencer module
     gosubEXECUTESEQ       'do sequence
.
.                          'Rest script body
gotoMENU                   'go to menu
'****As the CODE below allows the incorporation of the sequences in the
script program hence it must not be changed and should be added without
modification to the end of the script.****
:
LOADSEQUENCE::
 pushSEQUENCE%
 gosub"seq*,p",LOADSEQEXT
BRSEQDONE::
 pullSEQERROR%
 lprintSEQERROR%
 if(SEQERROR%=0)trueif
  SEQUENCE_LOAD%=-1
  RESULT$="SELOK:"
  a$=RESULT$+" sequence"+fnSTR$(SEQUENCE%)+" loaded"
 falseif
  SEQUENCE_LOAD%=0
  RESULT$="SELERROR:"
  a$=RESULT$+" in sequence"+fnSTR$(SEQUENCE%)
 endif
 lprinta$
return
:
EXECUTESEQ::
```

```
 gosubSET_SEQU_FILE
 gosub"seq*,p",DOSEQEXT
 pullSEQERROR%,SEQERR$
 if(SEQERROR%=0)trueif
  RESULT$="SEQOK:"
 falseif
  RESULT$="SEQERROR"+fnSTR$(SEQERROR%)+"  "+SEQERR$+":"
 endif
 lprintRESULT$
return
:
START::
 on(1-INIT%)gosubDEFAULTS
 INIT%=1
return
:
DEFAULTS::
 gosubPF
 gosub"seq*,p",INIT
 INIT%=1
return
:
SET_SEQU_FILE::
 SEQ_PATH$="@c:\thales\temp"
 SAVEFILE$="Sequence"+fnSTR$(SEQUENCE%)
 pushSEQ_PATH$,SAVEFILE$
 fnSEQ_EXE("pullREMO_ASCPATH$,REMO_ROOTFILE$")
return
:
deffnSEQ_EXE(a$,i,a)
 whileinstr(a$,"''")
  i=instr(a$,"''")
  a$=left$(a$,i-1)+chr$(34)+mid$(a$,i+2)
 wend
 a$=a$+":push0:return"
 pusha$
 gosub"seq*,p",EXECUTE
 pulla
fnenda
:
SCRIPT_END
```

7 Choosing different potentiostat

A power potentiostat or an electronic load can be chosen for high current/voltage application. To choose a power potentiostat or electronic load, click on “CONTROL POTENTIOSTAT”. This will open the TEST SAMPLING window.

In TEST SAMPLING window click on DEVICE. This will open a sub window, where the EPC42 port number must be given with which the power potentiostat or the electronic load is connected.