

“Timed sequence task” - Operant box code

Task design: Helena Janickova ( <https://orcid.org/0000-0003-0310-7893>)

Code author: Milan Janicek ( <https://orcid.org/0000-0002-8271-3674>)

Set of codes for standard operant boxes equipped with two fixed levers and a feeder (Med Associates, MED-307A-B1) and operated by Med-PC V software. The code is based on a code originally provided by MedAssociates (MEDLab8 Chain).

Names of the codes indicate the required sequences of lever presses. Individual inter-press intervals can be defined and easily modified and are logged in great detail (ie. different types of incorrect presses are distinguished and recorded separately).

Detailed description of the algorithms and all logged values can be found below.

Python scripts for the automated analysis of the resulting task logs are available on GitHub.

Code is available under FreeBSD license (<https://opensource.org/license/bsd-2-clause/>) .

Codes

[timed-LeftLeft.mpc](#)

[timed-LeftLeftRight.mpc](#)

[timed-LeftRight.mpc](#)

[timed-LeftLeftRightRight.mpc](#)

[timed-LeftLeftRightRight-noReinf.mpc](#)

[timed-LeftLeftRightRight-alternating.mpc](#)

Codes

timed-LeftLeft.mpc

what does it do?

- Sequence **LEFT - LEFT** is expected
- It is possible to set up timeouts
 - TimeoutLow to set up time mouse should wait until pressing 2nd LEFT
 - TimeoutHigh to set up time during which mouse should press 2nd LEFT
- there is short period of time after delivering reinforcement when mouse should wait before starting next round;
 - timeout is hardcoded to 5 seconds for unsuccessful trials
 - timeout is defined by TimeOut for successful trials

Sequence

- slightly simplified for readability - please check source code for all details
- (note that Sequences usually use different Z-Pulse signals so there is clear indication where mouse failed the task in the log)
- grey background describes the correct way through the task
- **S1** - start
- **S2** - waiting for first LEFT press
 - **LEFT**: correct, continue to **S3[low time state]**
 - **RIGHT** : incorrect, go to **S6[bad state]**
- **S3[low time state]** - mouse should wait for specified time (=TimeoutLow)
 - **LEFT**: incorrect (mouse should wait more..) go to **S6[bad state]**
 - **RIGHT** : incorrect, go to **S6[bad state]**
 - after time specified by TimeoutLow: automatically go to **S4[best time state]**
- **S4[best time state]**
 - **LEFT**: 2nd LEFT is pressed, go to **S10[reinforcement state]**
 - **RIGHT** : incorrect, go to **S6[bad state]**
 - after time specified by TimeoutHigh: automatically go to **S5[high time state]**
- **S5[high time state]**
 - **LEFT**: 2nd LEFT is pressed, but too late: go to **S6[bad state]**
 - **RIGHT** : incorrect, go to **S6[bad state]**
- **S6[bad state 1]**
 - immediately turn off lights, move to **S7[bad state 2]**
- **S7[bad state 2]** - during 5 seconds timeout
 - **LEFT**: log LEFT press during timeout, stay at **S7**
 - **RIGHT** : log RIGHT press during timeout, stay at **S7**
 - after 5 seconds: return to **S2**
- **S10[reinforcement state]** - deliver reinforcement and wait for time specified in TimeOut
 - **LEFT**: log LEFT press during timeout, stay at **S7**
 - **RIGHT** : log RIGHT press during timeout, stay at **S7**

- after time specified in TimeOut: return to **S2**

Following task specific codes are logged:

- .65/.50 = Incorrect Right
- .66/.50 = Incorrect Left (low time)
- .67/.50 = Incorrect Left (high time)
- .68/.50 = Incorrect - Press Left during timeout
- .69/.50 = Incorrect - Press Right during timeout

timed-LeftLeftRight.mpc

what does it do?

- Sequence **LEFT** - **LEFT** - **RIGHT** is expected
- It is possible to set up timeouts
 - TimeoutLowLeft to set up time mouse should wait until pressing 2nd LEFT
 - TimeoutHighLeft to set up time during which mouse should press 2nd LEFT
 - TimeoutLowRight to set up time mouse should wait until pressing 1st RIGHT
 - TimeoutHighRight to set up time during which mouse should press 1st RIGHT
- there is short period of time after delivering reinforcement when mouse should wait before starting next round;
 - timeout is hardcoded to 5 seconds for unsuccessful trials
 - timeout is defined by TimeOut for successful trials

Sequence

- slightly simplified for readability - please check source code for all details
- (note that Sequences usually use different Z-Pulse signals so there is clear indication where mouse failed the task in the log)
- grey background describes the correct way through the task
- **S1** - start
- **S2** - waiting for first LEFT press
 - **LEFT**: correct, continue to **S3[low time 2nd press state]**
 - **RIGHT**: incorrect, go to **S6[bad state]**
- **S3[low time 2nd press state]** - mouse should wait for specified time (=TimeoutLowLeft)
 - **LEFT**: incorrect (mouse should wait more..) go to **S6[bad state]**
 - **RIGHT**: incorrect, go to **S6[bad state]**
 - after time specified by TimeoutLowLeft: automatically go to **S4[best time state]**
- **S4[best time 2nd press state]**
 - **LEFT**: 2nd LEFT is pressed, go to **S23[low time 3rd press state]**
 - **RIGHT**: incorrect, go to **S6[bad state]**
 - after time specified by TimeoutHighLeft: automatically go to **S5[high time 2nd press state]**
- **S5[high time 2nd press state]**
 - **LEFT**: 2nd LEFT is pressed, but too late: go to **S6[bad state]**
 - **RIGHT**: incorrect, go to **S6[bad state]**
- **S23[low time 3rd press state]**
 - **RIGHT**: incorrect (mouse should wait more..) go to **S6[bad state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
 - after time specified by TimeoutLowLeft: automatically go to **S24[best time 3rd press state]**
- **S24[best time 3rd press state]**
 - **RIGHT**: RIGHT is pressed 3rd, go to **S10[reinforcement state]**
 - **LEFT**: incorrect, go to **S6[bad state]**

- after time specified by TimeoutHighRight: automatically go to **S25[high time 3rd press state]**
- **S25[high time 3rd press state]**
 - **RIGHT**: RIGHT is pressed 3rd, but too late: go to **S6[bad state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
- **S6[bad state 1]**
 - immediately turn off lights, move to **S7[bad state 2]**
- **S7[bad state 2]** - during 5 seconds timeout
 - **LEFT**: log LEFT press during timeout, stay at **S7**
 - **RIGHT**: log RIGHT press during timeout, stay at **S7**
 - after 5 seconds: return to **S2**
- **S10[reinforcement state]** - deliver reinforcement and wait for time specified in TimeOut
 - **LEFT**: log LEFT press during timeout, stay at **S7**
 - **RIGHT**: log RIGHT press during timeout, stay at **S7**
 - after time specified in TimeOut: return to **S2**

Following task specific codes are logged:

- .65/.50 = Incorrect Right
- .66/.50 = Incorrect Left (low time)
- .67/.50 = Incorrect Left (high time)
- .68/.50 = Incorrect - Press Left during timeout
- .69/.50 = Incorrect - Press Right during timeout
- .75/.50 = Incorrect Left
- .76/.50 = Incorrect Right (low time)
- .77/.50 = Incorrect Right (high time)

timed-LeftRight.mpc

what does it do?

- Sequence **LEFT** - **RIGHT** is expected
- It is possible to set up timeouts
 - TimeoutLowRight to set up time mouse should wait until pressing RIGHT
 - TimeoutHighRight to set up time during which mouse should press RIGHT
- there is short period of time after delivering reinforcement when mouse should wait before starting next round;
 - timeout is hardcoded to 5 seconds for unsuccessful trials
 - timeout is defined by TimeOut for successful trials
- It is modified version of LLR script (see [timed-LeftLeftRight.mpc](#)) with one modification - sequence skips 2nd LEFT

Sequence

- slightly simplified for readability - please check source code for all details
- (note that Sequences usually use different Z-Pulse signals so there is clear indication where mouse failed the task in the log)
- grey background describes the correct way through the task
- **S1** - start
- **S2** - waiting for first LEFT press
 - **LEFT**: correct, continue to **S23[low time 2nd press state]**
 - **RIGHT**: incorrect, go to **S6[bad state]**
- **S3** - not used
- **S4** - not used
- **S5** - not used
- **S23[low time 2nd press state]**
 - **RIGHT**: incorrect (mouse should wait more..) go to **S6[bad state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
 - after time specified by TimeoutLowLeft: automatically go to **S24[best time 2nd press state]**
- **S24[best time 2nd press state]**
 - **RIGHT**: RIGHT is pressed 2nd, go to **S10[reinforcement state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
 - after time specified by TimeoutHighRight: automatically go to **S25[high time 2nd press state]**
- **S25[high time 2nd press state]**
 - **RIGHT**: RIGHT is pressed 3rd, but too late: go to **S6[bad state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
- **S6[bad state 1]**
 - immediately turn off lights, move to **S7[bad state 2]**
- **S7[bad state 2]** - during 5 seconds timeout
 - **LEFT**: log LEFT press during timeout, stay at **S7**
 - **RIGHT**: log RIGHT press during timeout, stay at **S7**
 - after 5 seconds: return to **S2**

- ***S10[reinforcement state]*** - deliver reinforcement and wait for time specified in TimeOut
 - **LEFT**: log LEFT press during timeout, stay at **S7**
 - **RIGHT**: log RIGHT press during timeout, stay at **S7**
 - after time specified in TimeOut: return to **S2**

Following task specific codes are logged:

- .68/.50 = Incorrect - Press Left during timeout
- .69/.50 = Incorrect - Press Right during timeout
- .75/.50 = Incorrect Left
- .76/.50 = Incorrect Right (low time)
- .77/.50 = Incorrect Right (high time)

timed-LeftLeftRightRight.mpc

what does it do?

- Sequence **LEFT - LEFT - RIGHT - RIGHT** is expected
- It is possible to set up timeouts
 - TimeoutLowLeft to set up time mouse should wait until pressing 2nd LEFT
 - TimeoutHighLeft to set up time during which mouse should press 2nd LEFT
 - TimeoutLow1stRight to set up time mouse should wait until pressing 1st RIGHT
 - TimeoutHigh1stRight to set up time during which mouse should press 1st RIGHT
 - TimeoutLow2ndRight to set up time mouse should wait until pressing 2nd RIGHT
 - TimeoutHigh2ndRight to set up time during which mouse should press 2nd RIGHT
- there is short period of time after delivering reinforcement when mouse should wait before starting next round;
 - timeout is hardcoded to 5 seconds for unsuccessful trials
 - timeout is defined by TimeOut for successful trials

Sequence

- slightly simplified for readability - please check source code for all details
- (note that Sequences usually use different Z-Pulse signals so there is clear indication where mouse failed the task in the log)
- grey background describes the correct way through the task
- **S1** - start
- **S2** - waiting for 1st LEFT press
 - **LEFT**: correct, continue to **S3[low time 2nd press state]**
 - **RIGHT**: incorrect, go to **S6[bad state]**
- **S3[low time 2nd press state]** - mouse should wait for specified time (=TimeoutLowLeft)
 - **LEFT**: incorrect (mouse should wait more..) go to **S6[bad state]**
 - **RIGHT**: incorrect, go to **S6[bad state]**
 - after time specified by TimeoutLowLeft: automatically go to **S4[best time 2nd press state]**
- **S4[best time 2nd press state]**
 - **LEFT**: 2nd LEFT is pressed, go to **S23[low time 3rd press state]**
 - **RIGHT**: incorrect, go to **S6[bad state]**
 - after time specified by TimeoutHighLeft: automatically go to **S5[high time 2nd press state]**
- **S5[high time 2nd press state]**
 - **LEFT**: 2nd LEFT is pressed, but too late: go to **S6[bad state]**
 - **RIGHT**: incorrect, go to **S6[bad state]**
- **S23[low time 3rd press state]**

- **RIGHT**: incorrect (mouse should wait more..) go to **S6[bad state]**
- **LEFT**: incorrect, go to **S6[bad state]**
- after time specified by TimeoutLowLeft: automatically go to **S24[best time 3rd press state]**
- **S24[best time 3rd press state]**
 - **RIGHT**: RIGHT is pressed 3rd, go to **S26[low time 4th press state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
 - after time specified by TimeoutHighRight: automatically go to **S25[high time 3rd press state]**
- **S25[high time 3rd press state]**
 - **RIGHT**: RIGHT is pressed 3rd, but too late: go to **S6[bad state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
- **S26[low time 4th press state]**
 - **RIGHT**: incorrect (mouse should wait more..) go to **S6[bad state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
 - after time specified by TimeoutLowLeft: automatically go to **S27[best time 4th press state]**
- **S27[best time 4th press state]**
 - **RIGHT**: RIGHT is pressed 4th, go to **S10[reinforcement state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
 - after time specified by TimeoutHighRight: automatically go to **S28[high time 4th press state]**
- **S28[high time 4th press state]**
 - **RIGHT**: RIGHT is pressed 4th, but too late: go to **S6[bad state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
- **S6[bad state 1]**
 - immediately turn off lights, move to **S7[bad state 2]**
- **S7[bad state 2]** - during 5 seconds timeout
 - **LEFT**: log LEFT press during timeout, stay at **S7**
 - **RIGHT**: log RIGHT press during timeout, stay at **S7**
 - after 5 seconds: return to **S2**
- **S10[reinforcement state]** - deliver reinforcement and wait for time specified in TimeOut
 - **LEFT**: log LEFT press during timeout, stay at **S7**
 - **RIGHT**: log RIGHT press during timeout, stay at **S7**
 - after time specified in TimeOut: return to **S2**

timed-LeftLeftRightRight-noReinf.mpc

what does it do?

- Sequence **LEFT** - **LEFT** - **RIGHT** - **RIGHT** is expected
 - there is NO REINFORCEMENT delivered - only modification from [timed-LeftLeftRightRight.mpc](#)
 - reinforcement code is modified in “REINFORCEMENT TIMER” section
- It is possible to set up timeouts
 - TimeoutLowLeft to set up time mouse should wait until pressing 2nd LEFT
 - TimeoutHighLeft to set up time during which mouse should press 2nd LEFT
 - TimeoutLow1stRight to set up time mouse should wait until pressing 1st RIGHT
 - TimeoutHigh1stRight to set up time during which mouse should press 1st RIGHT
 - TimeoutLow2ndRight to set up time mouse should wait until pressing 2nd RIGHT
 - TimeoutHigh2ndRight to set up time during which mouse should press 2nd RIGHT
- there is short period of time after delivering reinforcement when mouse should wait before starting next round;
 - timeout is hardcoded to 5 seconds for unsuccessful trials
 - timeout is defined by TimeOut for successful trials

Sequence

- slightly simplified for readability - please check source code for all details
- (note that Sequences usually use different Z-Pulse signals so there is clear indication where mouse failed the task in the log)
- grey background describes the correct way through the task
- **S1** - start
- **S2** - waiting for 1st LEFT press
 - **LEFT**: correct, continue to **S3[low time 2nd press state]**
 - **RIGHT**: incorrect, go to **S6[bad state]**
- **S3[low time 2nd press state]** - mouse should wait for specified time (=TimeoutLowLeft)
 - **LEFT**: incorrect (mouse should wait more..) go to **S6[bad state]**
 - **RIGHT**: incorrect, go to **S6[bad state]**
 - after time specified by TimeoutLowLeft: automatically go to **S4[best time 2nd press state]**
- **S4[best time 2nd press state]**
 - **LEFT**: 2nd LEFT is pressed, go to **S23[low time 3rd press state]**
 - **RIGHT**: incorrect, go to **S6[bad state]**
 - after time specified by TimeoutHighLeft: automatically go to **S5[high time 2nd press state]**
- **S5[high time 2nd press state]**

- **LEFT**: 2nd LEFT is pressed, but too late: go to **S6[bad state]**
 - **RIGHT**: incorrect, go to **S6[bad state]**
- **S23[low time 3rd press state]**
 - **RIGHT**: incorrect (mouse should wait more..) go to **S6[bad state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
 - after time specified by TimeoutLowLeft: automatically go to **S24[best time 3rd press state]**
- **S24[best time 3rd press state]**
 - **RIGHT**: RIGHT is pressed 3rd, go to **S26[low time 4th press state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
 - after time specified by by TimeoutHighRight: automatically go to **S25[high time 3rd press state]**
- **S25[high time 3rd press state]**
 - **RIGHT**: RIGHT is pressed 3rd, but too late: go to **S6[bad state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
- **S26[low time 4th press state]**
 - **RIGHT**: incorrect (mouse should wait more..) go to **S6[bad state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
 - after time specified by TimeoutLowLeft: automatically go to **S27[best time 4th press state]**
- **S27[best time 4th press state]**
 - **RIGHT**: RIGHT is pressed 4th, go to **S10[reinforcement state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
 - after time specified by by TimeoutHighRight: automatically go to **S28[high time 4th press state]**
- **S28[high time 4th press state]**
 - **RIGHT**: RIGHT is pressed 4th, but too late: go to **S6[bad state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
- **S6[bad state 1]**
 - immediately turn off lights, move to **S7[bad state 2]**
- **S7[bad state 2]** - during 5 seconds timeout
 - **LEFT**: log LEFT press during timeout, stay at **S7**
 - **RIGHT**: log RIGHT press during timeout, stay at **S7**
 - after 5 seconds: return to **S2**
- **S10[reinforcement state]** - **DO NOT DELIVER REINFORCEMENT** and wait for time specified in TimeOut
 - **LEFT**: log LEFT press during timeout, stay at **S7**
 - **RIGHT**: log RIGHT press during timeout, stay at **S7**
 - after time specified in TimeOut: return to **S2**

timed-LeftLeftRightRight-alternating.mpc

what does it do?

- Sequence
 - **LEFT** - **LEFT** - **RIGHT** - **RIGHT** is expected in every 1st, 2nd and 3rd trial
 - **LEFT** - **LEFT** - **RIGHT** is expected every 4th trial
- based on [timed-LeftLeftRightRight.mpc](#) - mechanism was added to detect 4th trial
- It is possible to set up timeouts
 - TimeoutLowLeft to set up time mouse should wait until pressing 2nd LEFT
 - TimeoutHighLeft to set up time during which mouse should press 2nd LEFT
 - TimeoutLow1stRight to set up time mouse should wait until pressing 1st RIGHT
 - TimeoutHigh1stRight to set up time during which mouse should press 1st RIGHT
 - TimeoutLow2ndRight to set up time mouse should wait until pressing 2nd RIGHT
 - TimeoutHigh2ndRight to set up time during which mouse should press 2nd RIGHT
- there is short period of time after delivering reinforcement when mouse should wait before starting next round;
 - timeout is hardcoded to 5 seconds for unsuccessful trials
 - timeout is defined by TimeOut for successful trials

Sequence

- slightly simplified for readability - please check source code for all details
- (note that Sequences usually use different Z-Pulse signals so there is clear indication where mouse failed the task in the log)
- grey background describes the correct way through the task
- **S1** - start
- **S2** - waiting for 1st LEFT press
 - **LEFT**: correct, continue to **S3[low time 2nd press state]**
 - **RIGHT** : incorrect, go to **S6[bad state]**
- **S3[low time 2nd press state]** - mouse should wait for specified time (=TimeoutLowLeft)
 - **LEFT**: incorrect (mouse should wait more..) go to **S6[bad state]**
 - **RIGHT** : incorrect, go to **S6[bad state]**
 - after time specified by TimeoutLowLeft: automatically go to **S4[best time 2nd press state]**
- **S4[best time 2nd press state]**
 - **LEFT**: 2nd LEFT is pressed, go to **S23[low time 3rd press state]**
 - **RIGHT** : incorrect, go to **S6[bad state]**
 - after time specified by TimeoutHighLeft: automatically go to **S5[high time 2nd press state]**
- **S5[high time 2nd press state]**

- **LEFT**: 2nd LEFT is pressed, but too late: go to **S6[bad state]**
 - **RIGHT** : incorrect, go to **S6[bad state]**
- **S23[low time 3rd press state]**
 - **RIGHT** : incorrect (mouse should wait more..) go to **S6[bad state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
 - after time specified by TimeoutLowLeft: automatically go to **S24[best time 3rd press state]**
- **S24[best time 3rd press state]**
 - **RIGHT** : RIGHT is pressed 3rd, go to **S11[4th trial check state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
 - after time specified by by TimeoutHighRight: automatically go to **S25[high time 3rd press state]**
- **S25[high time 3rd press state]**
 - **RIGHT** : RIGHT is pressed 3rd, but too late: go to **S6[bad state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
- **S11[4th trial check state]**
 - check if trial counter (M) = 4
 - M == 4: we are in LEFT - LEFT - RIGHT scenario - go to **S10[reinforcement]**
 - M != 4 we are in LEFT - LEFT - RIGHT - RIGHT scenario - go to **S26[low time 4th press state]**
- **S26[low time 4th press state]**
 - **RIGHT** : incorrect (mouse should wait more..) go to **S6[bad state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
 - after time specified by TimeoutLowLeft: automatically go to **S27[best time 4th press state]**
- **S27[best time 4th press state]**
 - **RIGHT** : RIGHT is pressed 4th, go to **S10[reinforcement state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
 - after time specified by by TimeoutHighRight: automatically go to **S28[high time 4th press state]**
- **S28[high time 4th press state]**
 - **RIGHT** : RIGHT is pressed 4th, but too late: go to **S6[bad state]**
 - **LEFT**: incorrect, go to **S6[bad state]**
- **S6[bad state 1]**
 - immediately turn off lights, move to **S7[bad state 2]**
- **S7[bad state 2]** - during 5 seconds timeout
 - **LEFT**: log LEFT press during timeout, stay at **S7**
 - **RIGHT** : log RIGHT press during timeout, stay at **S7**
 - after 5 seconds: return to **S2**
- **S10[reinforcement state]** - deliver reinforcement and wait for time specified in TimeOut
 - **LEFT**: log LEFT press during timeout, stay at **S7**
 - **RIGHT** : log RIGHT press during timeout, stay at **S7**
 - after time specified in TimeOut: return to **S2**