



Feb 28, 2025

Non-human Primate Arm-reaching Behavioral Tasks

DOI

<https://dx.doi.org/10.17504/protocols.io.5jyl8d41rg2w/v1>

Damien Pittard^{1,2}, Adriana Galvan^{1,2}, Thomas Wichmann^{1,2}

¹Emory University; ²ASAP



Johnson Agniswamy

Emory University

Create & collaborate more with a free account

Edit and publish protocols, collaborate in communities, share insights through comments, and track progress with run records.

Create free account

OPEN  ACCESS



DOI: <https://dx.doi.org/10.17504/protocols.io.5jyl8d41rg2w/v1>

Protocol Citation: Damien Pittard, Adriana Galvan, Thomas Wichmann 2025. Non-human Primate Arm-reaching Behavioral Tasks. **protocols.io** <https://dx.doi.org/10.17504/protocols.io.5jyl8d41rg2w/v1>

License: This is an open access protocol distributed under the terms of the **Creative Commons Attribution License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Protocol status: Working

We use this protocol and it's working

Created: February 25, 2025



Last Modified: February 28, 2025

Protocol Integer ID: 123528

Keywords: ASAPCRN, evaluation of parkinsonian animal, parkinsonian animal, reaching behavioral task, behavioral testing, behavioral task, behavioral tasks this protocol, nhp, training, simple arm, human

Funders Acknowledgements:

ASAP

Grant ID: ASAP-020572

Abstract

This protocol describes procedures used for training and behavioral testing of non-human primates (NHPs) in simple arm-reaching behavioral tasks, used for evaluation of parkinsonian animals.

Guidelines

Introduction

Arm-reaching tasks are often used in NHPs to study their motor skills, to examine neuronal responses to behavioral events, and to study brain abnormalities in models of movement disorders. While monkeys have been shown to be able to learn highly complex motor tasks under healthy conditions, it has been difficult to study their performance in complex tasks in the parkinsonian state. Another limitation of behavioral testing in the parkinsonian state is that animals will perform even simple tasks only for a small number of trials. As we are interested in examining task responses under parkinsonian conditions, we chose a very simple task for our studies.

Materials

- Primate chair
- Head fixation via use of a thermoplastic molded helmet or a conventional head post
- Computer-controlled touchscreen display (GVision)

Safety warnings

! This protocol needs prior approval from the users' Institutional Animal Care and Use Committee (IACUC), the Institutional Review Board (IRB), or equivalent ethics committee.

Procedure/Setting

- 1 Train animals to accept being seated in a primate restraining chair, and to reduce their head movements through the use of a thermoplastic molded helmet (CIVCO medical solutions) (Drucker et al., 2015) or a surgically implanted head post.
- 2 Monkeys learn one of the arm-reaching tasks described below (implemented in MonkeyLogic v2.2 [NIMH]):
 - 2.1 Position the animals facing a 17-inch touchscreen display (GVision) and use their right arm to perform one of the tasks.

Note

The animals are not specifically food- or water-restricted but receive drops of preferred juice as an incentive to perform the task.

- 3 Consider training is complete once the animal consistently performs at least 100 successful trials per session.

Note

The physical aspects of the tasks are described in the following sections, while Matlab code and parameter file information is provided in the Appendix.

One-target Task

3s

- 4 Display a filled circle (~93 mm diameter) on the touch screen at randomly chosen intervals (see below), either at the center of the screen, or horizontally moved to either the left or right side of the screen (at the same vertical position).
- 5 The animal has to touch the circle to receive a juice reward.
- 6 At the moment of touch, the circle display will be switched off.
- 7 The target will be shown for a maximum of  00:00:03 .
 - If the monkey fails to touch the screen during this period, the circle display will be switched off (without reward).

3s



- 8 A new trial will be initiated after a random inter-trial interval of 2-3 seconds.
- If the animal has touched the screen, the inter-trial interval will start following the release of the monkey's hand from the touch screen.

Note

The timing and template parameters for this program are presented in Appendix 1.

Two-target Task

2s

- 9 This task requires the monkey to touch two separate targets in succession on the screen, with the arm contralateral to the recording site.
- 10 The trial begins with the display of a filled center target circle (~52 mm diameter). The monkey is required to touch the center target on the touch screen within  00:00:01 of its appearance.
- If the animal fails to touch the target, the trial is aborted (without reward).
- 11 If the monkey touches the center target, a similarly sized circular target on the left or right side of the screen (randomly chosen) appears, at an approximate distance of 117 mm (center – center) from the central target. The center target is extinguished when the side target appears.
- 12 Once the side target appears, the monkey has to touch the side target within  00:00:01 to receive a drop of juice for a successful trial.
- Failure to capture the peripheral target within the allotted time period aborts the trial.

1s

1s

Note

The timing and template parameters for this program are presented in Appendix 2.

APPENDIX

- 13 One-target task:

NIMH MonkeyLogic timing code (referred to as 'ag_6n_TTL' in the parameter file)

```
showcursor(false); % remove the touchscreen cursor
bhv_code(20, 'Sample', 40, 'Go', 50, 'Reward', 100, 'TrialEnd'); %
behavioral codes
sample = 1;
sound_sample = 2;
sample_time = 3000;
delay = 3000;
max_reaction_time = 1000;
itt_basic = 2000; % inter-trial intervals will not be any
shorter than this time (in ms)
itt_rand = 3000; % random component added to inter-trial
intervals (in ms)
hold_radius = 10; % fixation window (in degrees):

[~,button] = mouse_position();
while button(1) % if there is a mouse (=touch screen) input,
task is stuck
    [~,button] = mouse_position();
end
toggleobject([1 4], 'eventmarker', 20); % turn on the target
circle
ontarget =
eyejoytrack('touchtarget', sample, hold_radius, sample_time);
if ontarget
    toggleobject(3, 'status', 'on');
    idle(5); % duration of TTL pulse
    toggleobject(3, 'status', 'off');
    toggleobject(2); % turn on sound
    goodmonkey(300, 'juiceline', 1, 'numreward', 1,
'pausetime', 700, 'eventmarker', 50); % 300 ms
    toggleobject(2);
    trialerror(0); % good trial
else
    trialerror(3); % bad trial
end
toggleobject([1 4], 'eventmarker', 100);
idle(itt_basic + rand(1)*itt_rand); % wait ...
```

NIMH MonkeyLogic parameter information



	A	B	C	D	E	F	G	H
	Condi on	Freque ncy	Block	Timing File	TaskO bject#1	TaskO bject# 2	TaskObje ct#3	TaskObje ct#4
	1	1	1	ag_6n_ TTL	crc(5, [0 206 209],1,1 0,-5)	snd(sin ,0.5,10 00)	ttl(1)	ttl(2)
	2	1	1	ag_6n_ TTL	crc(5, [0 206 209],1, 0,-5)	snd(sin ,0.5,10 00)	ttl(1)	ttl(2)
	3	1	1	ag_6n_ TTL	crc(5, [0 206 209],1, -10,-5)	snd(sin ,0.5,10 00)	ttl(1)	ttl(2)

14 Two-target task:

NIMH MonkeyLogic timing code (referred to as 'Ulrik_2target') in the parameter file

```
bhv_code(15, 'CenterOn', 25, 'CenterOff', 30, 'CenterCapture', 40, 'CenterHold', 50, 'TargetOn', 60, 'TargetOff', 70, 'CenterLeave', 80, 'TargetCapture', 90, 'RewardOn', 100, 'RewardOff'); % behavioral codes

% **** User parameters (times in ms, fixation_window size in degrees) ****
% Object 1 (center):
acquisition_time1=1000; % max time for the monkey to catch the center (ms)
hold_time1_min =100; % min time for which the center has to be touched (ms)
hold_time1_max =100; % min time for which the center has to be touched (ms)
release_time1 =1000; % max duration before the monkey has to let go of the center
hold_radius1 = 3; % determines the precision with which the center has to be hit

% Object 2 (peripheral target):
acquisition_time2= 1000; % max time for the monkey to catch the center (ms)
hold_radius2 = 3; % size of the side target
reward_dur = 200; % duration of solenoid activation
intertrial_interval_min = 1500; % minimal intertrial interval duration
intertrial_interval_max = 1800; % maximal intertrial interval duration

%calculate randomized parameters
intertrial_interval = intertrial_interval_min + rand*(intertrial_interval_max - intertrial_interval_min);
hold_time1 = hold_time1_min + rand*(hold_time1_max - hold_time1_min);

toggleobject([1 2 3], 'status', 'off');
[~,button] = mouse_position();
while button(1) % if there is a mouse (=touch screen) input, task is stuck
    [~,button] = mouse_position();
end
toggleobject([1], 'status', 'on', 'eventmarker', 15); % turn center object on
on_center =
eyejoytrack('touchtarget', 1, hold_radius1, acquisition_time1);
```

```
if ~on_center % if center not hit within the appropriate time, the
trial is being terminated
    toggleobject([1 2 3], 'status', 'off', 'eventmarker', 25); %
switch center off
    trialerror(1); % report error type 1: did not touch center
    return
end
eventmarker(30); % set eventmarker: CenterCapture
center_hold =
eyejoytrack('releasetarget', 1, hold_radius1, hold_time1);
if ~center_hold % if the monkey released the center too early ...
    toggleobject([1 2 3], 'status', 'off', 'eventmarker', 25);
    trialerror(2); % report error type 2: broke center target
hold
    return
end
eventmarker(40); % set eventmarker: CenterHold
toggleobject([1], 'status', 'off', 'eventmarker', 25); % turn center
target off,
toggleobject([2], 'status', 'on', 'eventmarker', 50); % turn
peripheral target on (= GO signal)
[center_release, rt] =
eyejoytrack('releasetarget', 1, hold_radius1, release_time1);
if center_release % if monkey has not reacted within the
release_time1 window ...
    toggleobject([1 2 3], 'status', 'off', 'eventmarker', 60);
    trialerror(3); % report error type 3: did not react to
target
    return
end
eventmarker(70); % set eventmarker: CenterLeave
on_target =
eyejoytrack('touchtarget', 2, hold_radius2, acquisition_time2);
if ~on_target % if monkey has not acquired the peripheral target
within the time allowed ...
    toggleobject([1 2 3], 'status', 'off', 'eventmarker', 60);
    trialerror(4); % report error type 4: did not capture
target
    return
end
eventmarker(80); % set eventmarker: TargetCapture
trialerror(0); % report error type 0: GoodTrial
toggleobject(3, 'status', 'on'); % turn on sound
eventmarker(90); % set eventmarker: RewardOn
```



```
goodmonkey(200, 'juiceline',1, 'numreward',1, 'pausetime',500,
'eventmarker',80); % reward

eventmarker(60)
toggleobject([1 2 3],'status','off','eventmarker',100); % turn the
lights off
idle(intertrial_interval); % wait ...
```

NIMH MonkeyLogic parameter information

	A	B	C	D	E	F	G
	Condi on	Freque ncy	Bloc k	Timing File	TaskObject #1	TaskObject# 2	TaskObjec t#3
	1	1	1	Ulrik_2targ et	crc(3,[256 256 256],1,0,-5)	crc(3,[256 256 256],1,-13,-5)	snd(sin,0.5 ,1000)
	2	1	1	Ulrik_2targ et	crc(3,[256 256 256],1,0,-5)	crc(3,[256 256 256],1,13,-5)	snd(sin,0.5 ,1000)

Protocol references

Drucker, C.B., Carlson, M.L., Toda, K., DeWind, N.K., and Platt, M.L. (2015). Non-invasive primate head restraint using thermoplastic masks. *J Neurosci Methods* 253, 90-100.