

Examples

1. **mod** Python SDK Examples

Examples for AIRBOT Python SDK.

The examples are shipped with the AIRBOT Python SDK and can directly run on the command line. Running examples would require package `airbot_py` to be installed.

After launching the AIRBOT driver service (see step 1 and step 2 of [introduction](#)), the examples can be run with the following command:

```
python3 -m airbot_examples.<example_name> --port <port>
```

Available arguments for each examples could be examined by

```
python3 -m airbot_examples.<example_name> --help
```

Modules:

Name	Description
<code>app_load</code>	Example of loading an internal app
<code>app_unload</code>	Example of unloading an internal app
<code>get_end_pose</code>	Example of getting end effector pose.
<code>get_joint_pos</code>	Example of getting joint position.
<code>get_params</code>	Example of getting parameters from control service.
<code>mit_joint_integrated</code>	Example of sending continuous mit joint commands to the robot arm.
<code>move_cart_pose</code>	Example of moving the robot arm to a specified cartesian pose.
<code>move_joint_pos</code>	Example of moving the robot to a joint position
<code>move_with_cart_waypoints</code>	Example of moving the robot arm with waypoints.
<code>move_with_joint_waypoints</code>	Example of moving the robot arm with waypoints.

Name	Description
<code>servo_cart_pose</code>	Example of sending continuous cartesian pose commands to the robot arm.
<code>servo_cart_twist</code>	Example of sending continuous cartesian twist commands to the robot arm.
<code>servo_joint_pos</code>	Example of sending continuous joint position commands to the robot arm.
<code>set_params</code>	Example of setting parameters to control service.
<code>set_speed_profile</code>	Example of demonstrating the speed profile of the robot arm.
<code>switch_mode</code>	Example of switching control mode.
<code>task_follow</code>	Make one arm follow the movement of another arm.
<code>task_kbd_ctrl</code>	Example application of controlling the robot arm using keyboard input.
<code>task_swing</code>	Example application: making the robot arm swing in a sinusoidal motion in joint space.
<code>task_wipe</code>	Example application: making the robot arm swing in a sinusoidal motion in cartesian space.

1.1 `mod` `airbot_examples.app_load`

Example of loading an internal app

Functions:

Name	Description
<code>app_load</code>	Load app according to the given app name

1.1.1 `func` `airbot_examples.app_load.app_load()`

Load app according to the given app name

Source code in `airbot_examples/app_load.py`

```
21 def app_load():
22     """
23     Load app according to the given app name
24     """
25
26     import argparse
27
28     from airbot_py.arm import AIRBOTPlay
29
30     parser = argparse.ArgumentParser(description="Load app example")
31     parser.add_argument(
32         "-u",
33         "--url",
34         type=str,
35         default="localhost",
36         help="Port to connect to the server",
37     )
38     parser.add_argument(
39         "-p", "--port", type=int, default=50051, help="Port to connect to the server"
40     )
41     parser.add_argument(
42         "-a",
43         "--app_name",
44         type=str,
45         default="record_replay_app/record_replay_app:RecordReplayApp",
46         choices=["record_replay_app/record_replay_app:RecordReplayApp"],
47         help="Name of the app to load",
48     )
49
50     args = parser.parse_args()
51     with AIRBOTPlay(url=args.url, port=args.port) as robot:
52         robot.load_app("record_replay_app/record_replay_app:RecordReplayApp")
```

1.2 `mod` `airbot_examples.app_unload`

Example of unloading an internal app

Functions:

Name	Description
<code>app_unload</code>	Unload app according to the given app name

1.2.1 `func` `airbot_examples.app_unload.app_unload()`

Unload app according to the given app name

Source code in `airbot_examples/app_unload.py`

```
21 def app_unload():
22     """
23     Unload app according to the given app name
24     """
25     import argparse
26
27     from airbot_py.arm import AIRBOTPlay
28
29     parser = argparse.ArgumentParser(description="Unload app example")
30     parser.add_argument(
31         "-u",
32         "--url",
33         type=str,
34         default="localhost",
35         help="Port to connect to the server",
36     )
37     parser.add_argument(
38         "-p", "--port", type=int, default=50051, help="Port to connect to the server"
39     )
40     args = parser.parse_args()
41     with AIRBOTPlay(url=args.url, port=args.port) as robot:
42         robot.unload_app()
```

1.3 `mod` `airbot_examples.get_end_pose`

Example of getting end effector pose.

The end effector pose along with robot state and control mode will be printed.

Functions:

Name	Description
<code>get_end_pose</code>	Get end effector pose

1.3.1 `func` `airbot_examples.get_end_pose.get_end_pose()`

Get end effector pose

Source code in `airbot_examples/get_end_pose.py`



```

23 def get_end_pose():
24     """
25     Get end effector pose
26     """
27     import argparse
28     import time
29
30     from airbot_py.arm import AIRBOTPlay
31
32     parser = argparse.ArgumentParser(description="Get arm information example 2")
33     parser.add_argument(
34         "-f",
35         "--freq",
36         type=int,
37         default=10,
38         help="Frequency of printing joint positions",
39     )
40     parser.add_argument(
41         "-u",
42         "--url",
43         type=str,
44         default="localhost",
45         help="Port to connect to the server",
46     )
47     parser.add_argument(
48         "-p", "--port", type=int, default=50051, help="Port to connect to the server"
49     )
50     args = parser.parse_args()
51
52     idx = 0
53     with AIRBOTPlay(url=args.url, port=args.port) as robot:
54         product_info = robot.get_product_info()
55         print("-----")
56         print(f"Product name: {product_info['product_type']}")
57         print(f"Serial number: {product_info['sn']}")
58         print(f"Simulation mode: {product_info['is_sim']}")
59         print(f"Using interfaces: {product_info['interfaces']}")
60         print(f"Installed end effectors: {product_info['eef_types']}")
61
62         while True:
63             #####
64             # Get joint position and end effector position start
65             pose = robot.get_end_pose()
66             # Get joint position and end effector position end
67             #####
68             if idx % args.freq == 0:
69                 server_state = robot.get_state().name
70                 control_mode = robot.get_control_mode().name
71                 print("-----")
72                 print(f"Server state: {server_state}")
73                 print(f"Control mode: {control_mode}")
74                 print(
75                     "".join(
76                         [f"{i:>8s}" for i in ["x", "y", "z", "qx", "qy", "qz", "qw"]]
77                     )
78                 )
79                 print("-----")
80             print(
81                 (
82                     "".join([f"{i:+8.2f}" for i in pose[0]])
83                     if pose is not None
84                     else "Not received yet"
85                 ),
86                 end="",

```

```

    )
    print("".join([f"{i:+8.2f}" for i in pose[1]]) if pose is not None else "")
    time.sleep(1 / args.freq)
    idx += 1

```

```

86
87
88
89
90

```

1.4 `mod` `airbot_examples.get_joint_pos`

Example of getting joint position.

The joint position along with robot state and control mode will be printed.

Functions:

Name	Description
<code>get_joint_pos</code>	Get joint position

1.4.1 `func` `airbot_examples.get_joint_pos.get_joint_pos()`

Get joint position

Source code in `airbot_examples/get_joint_pos.py`



```

23 def get_joint_pos():
24     """
25     Get joint position
26     """
27     import argparse
28     import math
29     import time
30
31     from airbot_py.arm import AIRBOTPlay
32
33     parser = argparse.ArgumentParser(description="Get arm information example 1")
34     parser.add_argument(
35         "-f",
36         "--freq",
37         type=int,
38         default=10,
39         help="Frequency of printing joint positions",
40     )
41     parser.add_argument(
42         "-u",
43         "--url",
44         type=str,
45         default="localhost",
46         help="Port to connect to the server",
47     )
48     parser.add_argument(
49         "-p", "--port", type=int, default=50051, help="Port to connect to the server"
50     )
51     parser.add_argument(
52         "--radian",
53         action="store_true",
54         help="Print joint positions in radians instead of degrees",
55     )
56     args = parser.parse_args()
57
58     idx = 0
59     with AIRBOTPlay(url=args.url, port=args.port) as robot:
60         product_info = robot.get_product_info()
61         print("-----")
62         print(f"Product name: {product_info['product_type']}")
63         print(f"Serial number: {product_info['sn']}")
64         print(f"Simulation mode: {product_info['is_sim']}")
65         print(f"Using interfaces: {product_info['interfaces']}")
66         print(f"Installed end effectors: {product_info['eef_types']}")
67         print(f"Firmware versions: {product_info['fw_versions']}")
68         print("-----")
69
70         while True:
71             #####
72             # Get joint position and end effector position start
73             pos = robot.get_joint_pos()
74             eef_pos = robot.get_eef_pos()
75             # Get joint position and end effector position end
76             #####
77             if idx % args.freq == 0:
78                 server_state = robot.get_state().name
79                 control_mode = robot.get_control_mode().name
80                 print(
81                     "-----"
82                 )
83                 print(f"Server state: {server_state}")
84                 print(f"Control mode: {control_mode}")
85                 print(
86                     " ".join([f"{'joint':>7s}{i+1}" for i in range(6)])

```

```

        + f"{'end eef':>10s}"
    )
    print(
        "-----"
    )
    print(
        (
            ", ".join(
                [
                    (
                        f"{i / math.pi * 180:+7.2f}°"
                        if not args.radian
                        else f"{i:7.2f}"
                    )
                    for i in pos
                ]
            )
            if pos is not None
            else "None received yet"
        ),
        end="",
    )
    print(
        f"{eef_pos[0] * 100:8.2f} cm"
        if eef_pos is not None and len(eef_pos) > 0
        else ""
    )
    time.sleep(1 / args.freq)
    idx += 1

```

86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115

Example of getting parameters from control service.

Functions:

Name	Description
<code>get_params</code>	Example of getting parameters from control service.

1.5.1 `func` `airbot_examples.get_params.get_params()`

Example of getting parameters from control service.

Source code in `airbot_examples/get_params.py`

```
21 def get_params():
22     """
23     Example of getting parameters from control service.
24     """
25     import argparse
26
27     from airbot_py.arm import AIRBOTPlay
28
29     parser = argparse.ArgumentParser(description="Get params example")
30     parser.add_argument(
31         "-u",
32         "--url",
33         type=str,
34         default="localhost",
35         help="Port to connect to the server",
36     )
37     parser.add_argument(
38         "-p", "--port", type=int, default=50051, help="Port to connect to the server"
39     )
40     parser.add_argument(
41         "params",
42         action="append",
43         default=[],
44         help="Parameters to get",
45     )
46
47     args = parser.parse_args()
48
49     with AIRBOTPlay(url=args.url, port=args.port) as robot:
50         print(robot.get_params(args.params))
```

1.6 `mod` `airbot_examples.mit_joint_integrated`

Example of sending continuous mit joint commands to the robot arm.

Functions:

Name	Description
------	-------------

Name	Description
<code>mit_joint_integrated</code>	Example of sending continuous joint position commands to the robot arm.

1.6.1 `func` `airbot_examples.mit_joint_integrated.mit_joint_integrated()`

Example of sending continuous joint position commands to the robot arm.

Source code in `airbot_examples/mit_joint_integrated.py`



```

21 def mit_joint_integrated():
22     """
23     Example of sending continuous joint position commands to the robot arm.
24     """
25
26     import argparse
27     import json
28     import time
29
30     from airbot_py.arm import AIRBOTPlay, RobotMode, SpeedProfile
31
32     print(
33         "\x1b[1;33mThe robot arm is about to be move. Please make sure no obstacles exist in
34 the surrounding environment.\x1b[0m"
35     )
36     input("\x1b[1;35mPress Enter to continue...\x1b[0m")
37     print("\x1b[1;32mPress Ctrl+C to stop...\x1b[0m")
38
39     parser = argparse.ArgumentParser(description="Servo example")
40     parser.add_argument(
41         "-u",
42         "--url",
43         type=str,
44         default="localhost",
45         help="Port to connect to the server",
46     )
47     parser.add_argument(
48         "-p", "--port", type=int, default=50051, help="Port to connect to the server"
49     )
50     parser.add_argument(
51         "--pos",
52         type=json.loads,
53         help="Target joint positions, a 6-float array",
54         default=[0.0, -1.19, 1.17, -1.55, 1.51, -0.00],
55     )
56     parser.add_argument(
57         "--vel",
58         type=json.loads,
59         help="Target joint velocities, a 6-float array",
60         default=[0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
61     )
62     parser.add_argument(
63         "--eff",
64         type=json.loads,
65         help="Target joint efforts, a 6-float array",
66         default=[0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
67     )
68     parser.add_argument(
69         "--kp",
70         type=json.loads,
71         help="Target joint kps, a 6-float array",
72         default=[90.0, 150.0, 150.0, 25.0, 25.0, 25.0],
73     )
74     parser.add_argument(
75         "--kd",
76         type=json.loads,
77         help="Target joint kds, a 6-float array",
78         default=[2.5, 1.75, 1.5, 0.5, 1.5, 0.5],
79     )
80     parser.add_argument(
81         "-S",
82         "--speed",
83         choices=["slow", "fast", "default"],
84         default="default",

```

```

        help="Speed profile for the robot arm",
    )
    args = parser.parse_args()
    with AIRBOTPlay(url=args.url, port=args.port) as robot:
        if args.speed == "slow":
            robot.set_speed_profile(SpeedProfile.SLOW)
        elif args.speed == "fast":
            robot.set_speed_profile(SpeedProfile.FAST)
        else:
            robot.set_speed_profile(SpeedProfile.DEFAULT)
        robot.switch_mode(RobotMode.PLANNING_POS)
        try:
            pos = [float(i) for i in args.pos]
            vel = [float(i) for i in args.vel]
            eff = [float(i) for i in args.eff]
            kp = [float(i) for i in args.kp]
            kd = [float(i) for i in args.kd]
            # time.sleep(3)
            robot.move_to_joint_pos(pos)
            robot.switch_mode(RobotMode.MIT_INTEGRATED)
            import time

            joint1_upper = 1.57
            joint1_lower = -1.57
            delta_pos = 0.001 * 1.5
            direction = 1

            while True:
                robot.mit_joint_integrated_control(pos, vel, eff, kp, kd)
                if direction == 1:
                    pos[0] += delta_pos
                    if pos[0] >= joint1_upper:
                        pos[0] = joint1_upper
                        direction = -1
                else:
                    pos[0] -= delta_pos
                    if pos[0] <= joint1_lower:
                        pos[0] = joint1_lower
                        direction = 1
                time.sleep(0.8 / 250)
            finally:
                robot.set_speed_profile(SpeedProfile.DEFAULT)

```

84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105

```
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
```

1.7 **mod** `airbot_examples.move_cart_pose`

Example of moving the robot arm to a specified cartesian pose.

Functions:

Name	Description
<code>move_cart_pose</code>	Example of moving the robot arm to a specified cartesian pose.

1.7.1 **func** `airbot_examples.move_cart_pose.move_cart_pose()`

Example of moving the robot arm to a specified cartesian pose.

Source code in `airbot_examples/move_cart_pose.py`



```

21 def move_cart_pose():
22     """
23     Example of moving the robot arm to a specified cartesian pose.
24     """
25     import argparse
26     import json
27
28     print(
29         "\x1b[1;33mThe robot arm is about to be move. Please make sure no obstacles exist in
30 the surrounding environment.\x1b[0m"
31     )
32     input("\x1b[1;35mPress Enter to continue...\x1b[0m")
33
34     from airbot_py.arm import AIRBOTPlay, RobotMode, SpeedProfile
35
36     parser = argparse.ArgumentParser(description="Servo example")
37     parser.add_argument(
38         "-u",
39         "--url",
40         type=str,
41         default="localhost",
42         help="Port to connect to the server",
43     )
44     parser.add_argument(
45         "-p", "--port", type=int, default=50051, help="Port to connect to the server"
46     )
47     parser.add_argument(
48         "-t",
49         "--translation",
50         type=json.loads,
51         help="Target translation in the cartesian space, a 3-float array",
52         default=[0.45, 0.20, 0.45],
53     )
54     parser.add_argument(
55         "-r",
56         "--orientation",
57         type=json.loads,
58         help="Target orientation (in quaternion) in the cartesian space, a 4-float array",
59         default=[0.0, 0.0, 0.0, 1.0],
60     )
61     parser.add_argument(
62         "-e",
63         "--end-pos",
64         type=float,
65         help="Target end effector position, a float",
66         default=0.0,
67     )
68     parser.add_argument(
69         "-S",
70         "--speed",
71         choices=["slow", "fast", "default"],
72         default="default",
73         help="Speed profile for the robot arm",
74     )
75
76     args = parser.parse_args()
77     with AIRBOTPlay(url=args.url, port=args.port) as robot:
78         if args.speed == "slow":
79             robot.set_speed_profile(SpeedProfile.SLOW)
80         elif args.speed == "fast":
81             robot.set_speed_profile(SpeedProfile.FAST)
82         else:
83             robot.set_speed_profile(SpeedProfile.DEFAULT)
84         robot.switch_mode(RobotMode.PLANNING_POS)

```

```
robot.move_eef_pos([args.end_pos])
robot.move_to_cart_pose([args.translation, args.orientation])
robot.set_speed_profile(SpeedProfile.DEFAULT)
```

```
84
85
86
```

1.8 `mod` `airbot_examples.move_joint_pos`

Example of moving the robot to a joint position

Functions:

Name	Description
<code>move_joint_pos</code>	Move the robot to a joint position

1.8.1 `func` `airbot_examples.move_joint_pos.move_joint_pos()`

Move the robot to a joint position

```

21 def move_joint_pos():
22     """
23     Move the robot to a joint position
24     """
25     import argparse
26     import json
27
28     from airbot_py.arm import AIRBOTPlay, RobotMode, SpeedProfile
29
30     parser = argparse.ArgumentParser(description="Servo example")
31     parser.add_argument(
32         "-u",
33         "--url",
34         type=str,
35         default="localhost",
36         help="Port to connect to the server",
37     )
38     parser.add_argument(
39         "-p", "--port", type=int, default=50051, help="Port to connect to the server"
40     )
41     parser.add_argument(
42         "-j",
43         "--joints",
44         type=json.loads,
45         help="Target joint positions, a 6-float array",
46         default=[0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
47     )
48     parser.add_argument(
49         "-e",
50         "--end-pos",
51         type=float,
52         help="Target end effector position, a float",
53         default=0.0,
54     )
55     parser.add_argument(
56         "-S",
57         "--speed",
58         choices=["slow", "fast", "default"],
59         default="default",
60         help="Speed profile for the robot arm",
61     )
62
63     args = parser.parse_args()
64
65     with AIRBOTPlay(url=args.url, port=args.port) as robot:
66         if args.speed == "slow":
67             robot.set_speed_profile(SpeedProfile.SLOW)
68         elif args.speed == "fast":
69             robot.set_speed_profile(SpeedProfile.FAST)
70         else:
71             robot.set_speed_profile(SpeedProfile.DEFAULT)
72         robot.switch_mode(RobotMode.PLANNING_POS)
73         robot.move_eef_pos([args.end_pos])
74         robot.move_to_joint_pos([float(i) for i in args.joints])
75         robot.set_speed_profile(SpeedProfile.DEFAULT)

```

Example of moving the robot arm with waypoints.

Functions:

Name	Description
<code>move_with_cart_waypoints</code>	Example of moving the robot arm with waypoints..

1.9.1 `func` `airbot_examples.move_with_cart_waypoints.move_with_cart_waypoints()`

Example of moving the robot arm with waypoints..

Source code in `airbot_examples/move_with_cart_waypoints.py`



```

6  def move_with_cart_waypoints():
7      """
8      Example of moving the robot arm with waypoints..
9      """
10     import argparse
11     import json
12
13     print(
14         "\x1b[1;33mThe robot arm is about to be move. Please make sure no obstacles exist in
15 the surrounding environment.\x1b[0m"
16     )
17     input("\x1b[1;35mPress Enter to continue...\x1b[0m")
18
19     from airbot_py.arm import AIRBOTPlay, RobotMode, SpeedProfile
20
21     parser = argparse.ArgumentParser(description="Waypoints example")
22     parser.add_argument(
23         "-u",
24         "--url",
25         type=str,
26         default="localhost",
27         help="Port to connect to the server",
28     )
29     parser.add_argument(
30         "-p", "--port", type=int, default=50051, help="Port to connect to the server"
31     )
32     parser.add_argument(
33         "-w",
34         "--cart-waypoints",
35         type=json.loads,
36         help="Target waypoints as an array",
37         default=[
38             [[0.46, 0.0, 0.36], [0.0, 0.0, 0.0, 1.0]],
39             [[0.46, 0.1, 0.26], [0.0, 0.0, 0.0, 1.0]],
40             [[0.46, 0.0, 0.16], [0.0, 0.0, 0.0, 1.0]],
41             [[0.46, -0.1, 0.26], [0.0, 0.0, 0.0, 1.0]],
42         ],
43     )
44     parser.add_argument(
45         "-S",
46         "--speed",
47         choices=["slow", "fast", "default"],
48         default="default",
49         help="Speed profile for the robot arm",
50     )
51
52     parser.add_argument(
53         "--path",
54         action="store_true",
55         help="Execute path without planning",
56     )
57
58     args = parser.parse_args()
59     with AIRBOTPlay(url=args.url, port=args.port) as robot:
60         if args.speed == "slow":
61             robot.set_speed_profile(SpeedProfile.SLOW)
62         elif args.speed == "fast":
63             robot.set_speed_profile(SpeedProfile.FAST)
64         else:
65             robot.set_speed_profile(SpeedProfile.DEFAULT)
66         if args.path:
67             robot.switch_mode(RobotMode.PLANNING_WAYPOINTS_PATH)
68         else:
69             robot.switch_mode(RobotMode.PLANNING_WAYPOINTS)

```

```
robot.move_with_cart_waypoints(args.cart_waypoints)
robot.set_speed_profile(SpeedProfile.DEFAULT)
```

```
69
70
```

1.10 **mod** `airbot_examples.move_with_joint_waypoints`

Example of moving the robot arm with waypoints.

Functions:

Name	Description
<code>move_with_joint_waypoints</code>	Example of moving the robot arm with waypoints..

1.10.1 **func** `airbot_examples.move_with_joint_waypoints.move_with_joint_waypoints()`

Example of moving the robot arm with waypoints..

```

6  def move_with_joint_waypoints():
7      """
8      Example of moving the robot arm with waypoints..
9      """
10     import argparse
11     import json
12
13     print(
14         "\x1b[1;33mThe robot arm is about to be move. Please make sure no obstacles exist in
15 the surrounding environment.\x1b[0m"
16     )
17     input("\x1b[1;35mPress Enter to continue...\x1b[0m")
18
19     from airbot_py.arm import AIRBOTPlay, RobotMode, SpeedProfile
20
21     parser = argparse.ArgumentParser(description="Waypoints example")
22     parser.add_argument(
23         "-u",
24         "--url",
25         type=str,
26         default="localhost",
27         help="Port to connect to the server",
28     )
29     parser.add_argument(
30         "-p", "--port", type=int, default=50051, help="Port to connect to the server"
31     )
32     parser.add_argument(
33         "-w",
34         "--joint-waypoints",
35         type=json.loads,
36         help="Target translation in the cartesian space, a 3-float array",
37         default=[
38             [0.37, -0.6, 0.5, 0.132, -0.30, -0.13],
39             [0.2, -2.0, 2.0, 0.0, 0.01, 0.0],
40             [0.2, 0.1, 0.3, 0.0, 0.01, 0.0],
41             [-0.2, 0.15, 0.0, 0.0, 0.01, 0.0],
42         ],
43     )
44     parser.add_argument(
45         "-S",
46         "--speed",
47         choices=["slow", "fast", "default"],
48         default="default",
49         help="Speed profile for the robot arm",
50     )
51
52     args = parser.parse_args()
53     with AIRBOTPlay(url=args.url, port=args.port) as robot:
54         if args.speed == "slow":
55             robot.set_speed_profile(SpeedProfile.SLOW)
56         elif args.speed == "fast":
57             robot.set_speed_profile(SpeedProfile.FAST)
58         else:
59             robot.set_speed_profile(SpeedProfile.DEFAULT)
60         robot.switch_mode(RobotMode.PLANNING_WAYPOINTS)
61         robot.move_with_joint_waypoints(args.joint_waypoints)
62         robot.set_speed_profile(SpeedProfile.DEFAULT)

```

1.11 **mod** `airbot_examples.servo_cart_pose`

Example of sending continuous cartesian pose commands to the robot arm.

Functions:

Name	Description
<code>servo_cart_pose</code>	Example of sending continuous cartesian pose commands to the robot arm.

1.11.1 **func** `airbot_examples.servo_cart_pose.servo_cart_pose()`

Example of sending continuous cartesian pose commands to the robot arm.

Source code in `airbot_examples/servo_cart_pose.py`



```

21 def servo_cart_pose():
22     """
23     Example of sending continuous cartesian pose commands to the robot arm.
24     """
25
26     import argparse
27     import json
28     import time
29
30     print(
31         "\x1b[1;33mThe robot arm is about to be move. Please make sure no obstacles exist in
32 the surrounding environment.\x1b[0m"
33     )
34     input("\x1b[1;35mPress Enter to continue...\x1b[0m")
35     print("\x1b[1;32mPress Ctrl+C to stop...\x1b[0m")
36
37     from airbot_py.arm import AIRBOTPlay, RobotMode, SpeedProfile
38
39     parser = argparse.ArgumentParser(description="Servo example")
40     parser.add_argument(
41         "-u",
42         "--url",
43         type=str,
44         default="localhost",
45         help="Port to connect to the server",
46     )
47     parser.add_argument(
48         "-p", "--port", type=int, default=50051, help="Port to connect to the server"
49     )
50     parser.add_argument(
51         "-t",
52         "--translation",
53         type=json.loads,
54         help="Target translation in the cartesian space, a 3-float array",
55         default=[0.45, 0.20, 0.45],
56     )
57     parser.add_argument(
58         "-r",
59         "--orientation",
60         type=json.loads,
61         help="Target orientation (in quaternions) in the cartesian space, a 4-float array",
62         default=[0.0, 0.0, 0.0, 1.0],
63     )
64     parser.add_argument(
65         "-e",
66         "--end-pos",
67         type=float,
68         help="Target end effector position, a float",
69         default=0.0,
70     )
71     parser.add_argument(
72         "-s",
73         "--speed",
74         choices=["slow", "fast", "default"],
75         default="default",
76         help="Speed profile for the robot arm",
77     )
78
79     args = parser.parse_args()
80     with AIRBOTPlay(url=args.url, port=args.port) as robot:
81         if args.speed == "slow":
82             robot.set_speed_profile(SpeedProfile.SLOW)
83         elif args.speed == "fast":
84             robot.set_speed_profile(SpeedProfile.FAST)

```

```

else:
    robot.set_speed_profile(SpeedProfile.DEFAULT)
robot.switch_mode(RobotMode.SERVO_CART_POSE)
try:
    while True:
        robot.servo_cart_pose([args.translation, args.orientation])
        robot.servo_eef_pos([args.end_pos])
        time.sleep(0.01)
finally:
    robot.set_speed_profile(SpeedProfile.DEFAULT)

```

```

84
85
86
87
88
89
90
91
92
93

```

1.12 `mod` `airbot_examples.servo_cart_twist`

Example of sending continuous cartesian twist commands to the robot arm.

Functions:

Name	Description
<code>servo_cart_twist</code>	Example of sending continuous cartesian twist commands to the robot arm.

1.12.1 `func` `airbot_examples.servo_cart_twist.servo_cart_twist()`

Example of sending continuous cartesian twist commands to the robot arm.

Source code in `airbot_examples/servo_cart_twist.py`



```

21 def servo_cart_twist():
22     """
23     Example of sending continuous cartesian twist commands to the robot arm.
24     """
25
26     import argparse
27     import json
28     import time
29
30     print(
31         "\x1b[1;33mThe robot arm is about to be move. Please make sure no obstacles exist in
32 the surrounding environment.\x1b[0m"
33     )
34     input("\x1b[1;35mPress Enter to continue...\x1b[0m")
35     print("\x1b[1;32mPress Ctrl+C to stop...\x1b[0m")
36
37     from airbot_py.arm import AIRBOTPlay, RobotMode, SpeedProfile
38
39     parser = argparse.ArgumentParser(description="Servo example")
40     parser.add_argument(
41         "-u",
42         "--url",
43         type=str,
44         default="localhost",
45         help="Port to connect to the server",
46     )
47     parser.add_argument(
48         "-p", "--port", type=int, default=50051, help="Port to connect to the server"
49     )
50     parser.add_argument(
51         "-t",
52         "--translation",
53         type=json.loads,
54         help="Target translation in the cartesian space, a 3-float array",
55         default=[1.0, 0.0, 1.0],
56     )
57     parser.add_argument(
58         "-r",
59         "--rotation",
60         type=json.loads,
61         help="Target rotation in the cartesian space, a 3-float array",
62         default=[0.0, 0.0, 0.0],
63     )
64     parser.add_argument(
65         "-S",
66         "--speed",
67         choices=["slow", "fast", "default"],
68         default="default",
69         help="Speed profile for the robot arm",
70     )
71
72     args = parser.parse_args()
73     with AIRBOTPlay(url=args.url, port=args.port) as robot:
74         if args.speed == "slow":
75             robot.set_speed_profile(SpeedProfile.SLOW)
76         elif args.speed == "fast":
77             robot.set_speed_profile(SpeedProfile.FAST)
78         else:
79             robot.set_speed_profile(SpeedProfile.DEFAULT)
80         robot.switch_mode(RobotMode.SERVO_CART_TWIST)
81         try:
82             while True:
83                 robot.servo_cart_twist([args.translation, args.rotation])
84                 time.sleep(0.01)

```

```
finally:  
    robot.set_speed_profile(SpeedProfile.DEFAULT)
```

84

85

1.13 `mod` `airbot_examples.servo_joint_pos`

Example of sending continuous joint position commands to the robot arm.

Functions:

Name	Description
<code>servo_joint_pos</code>	Example of sending continuous joint position commands to the robot arm.

1.13.1 `func` `airbot_examples.servo_joint_pos.servo_joint_pos()`

Example of sending continuous joint position commands to the robot arm.

Source code in `airbot_examples/servo_joint_pos.py`



```

21 def servo_joint_pos():
22     """
23     Example of sending continuous joint position commands to the robot arm.
24     """
25
26     import argparse
27     import json
28     import time
29
30     from airbot_py.arm import AIRBOTPlay, RobotMode, SpeedProfile
31
32     print(
33         "\x1b[1;33mThe robot arm is about to be move. Please make sure no obstacles exist in
34 the surrounding environment.\x1b[0m"
35     )
36     input("\x1b[1;35mPress Enter to continue...\x1b[0m")
37     print("\x1b[1;32mPress Ctrl+C to stop...\x1b[0m")
38
39     parser = argparse.ArgumentParser(description="Servo example")
40     parser.add_argument(
41         "-u",
42         "--url",
43         type=str,
44         default="localhost",
45         help="Port to connect to the server",
46     )
47     parser.add_argument(
48         "-p", "--port", type=int, default=50051, help="Port to connect to the server"
49     )
50     parser.add_argument(
51         "-j",
52         "--joints",
53         type=json.loads,
54         help="Target joint positions, a 6-float array",
55         default=[0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
56     )
57
58     parser.add_argument(
59         "-e",
60         "--end-pos",
61         type=float,
62         help="Target end effector position, a float",
63         default=0.0,
64     )
65     parser.add_argument(
66         "-S",
67         "--speed",
68         choices=["slow", "fast", "default"],
69         default="default",
70         help="Speed profile for the robot arm",
71     )
72     args = parser.parse_args()
73     with AIRBOTPlay(url=args.url, port=args.port) as robot:
74         if args.speed == "slow":
75             robot.set_speed_profile(SpeedProfile.SLOW)
76         elif args.speed == "fast":
77             robot.set_speed_profile(SpeedProfile.FAST)
78         else:
79             robot.set_speed_profile(SpeedProfile.DEFAULT)
80         robot.switch_mode(RobotMode.SERVO_JOINT_POS)
81         try:
82             while True:
83                 pos = [float(i) for i in args.joints]
84                 import math

```

```

        pos[4] += math.sin(time.time_ns() / 1e9 * 2 * math.pi / 2)
        robot.servo_joint_pos(pos)
        robot.servo_eef_pos([args.end_pos])
        time.sleep(0.01)
    finally:
        robot.set_speed_profile(SpeedProfile.DEFAULT)

```

```

84
85
86
87
88
89
90

```

1.14 `mod` `airbot_examples.set_params`

Example of setting parameters to control service.

Functions:

Name	Description
<code>set_params</code>	Example of setting parameters to control service.

1.14.1 `func` `airbot_examples.set_params.set_params()`

Example of setting parameters to control service.

Source code in `airbot_examples/set_params.py`

```
21 def set_params():
22     """
23     Example of setting parameters to control service.
24     """
25     import argparse
26     import json
27
28     from airbot_py.arm import AIRBOTPlay
29
30     parser = argparse.ArgumentParser(description="Set params example")
31     parser.add_argument(
32         "-u",
33         "--url",
34         type=str,
35         default="localhost",
36         help="Port to connect to the server",
37     )
38     parser.add_argument(
39         "-p", "--port", type=int, default=50051, help="Port to connect to the server"
40     )
41     parser.add_argument(
42         "params",
43         default={},
44         type=json.loads,
45         help="Parameters to get, in json format.",
46     )
47
48     args = parser.parse_args()
49
50     with AIRBOTPlay(url=args.url, port=args.port) as robot:
51         robot.set_params(args.params)
```

1.15 `mod` `airbot_examples.set_speed_profile`

Example of demonstrating the speed profile of the robot arm.

Functions:

Name	Description
<code>set_speed_profile</code>	Example of demonstrating the speed profile of the robot arm.

1.15.1 `func` `airbot_examples.set_speed_profile.set_speed_profile()`

Example of demonstrating the speed profile of the robot arm.

```

21 def set_speed_profile():
22     """
23     Example of demonstrating the speed profile of the robot arm.
24     """
25
26     import argparse
27     import time
28
29     from airbot_py.arm import AIRBOTPlay, RobotMode, SpeedProfile
30
31     print(
32         "\x1b[1;33mThe robot arm is about to be move. Please make sure no obstacles exist in
33 the surrounding environment.\x1b[0m"
34     )
35     input("\x1b[1;35mPress Enter to continue...\x1b[0m")
36     print("\x1b[1;32mPress Ctrl+C to stop...\x1b[0m")
37
38     parser = argparse.ArgumentParser(description="Servo example")
39     parser.add_argument(
40         "-u",
41         "--url",
42         type=str,
43         default="localhost",
44         help="Port to connect to the server",
45     )
46     parser.add_argument(
47         "-p", "--port", type=int, default=50051, help="Port to connect to the server"
48     )
49     parser.add_argument(
50         "-S",
51         "--speed",
52         choices=["slow", "fast", "default"],
53         default="default",
54         help="Speed profile for the robot arm",
55     )
56     args = parser.parse_args()
57     with AIRBOTPlay(url=args.url, port=args.port) as robot:
58         if args.speed == "slow":
59             robot.set_speed_profile(SpeedProfile.SLOW)
60         elif args.speed == "fast":
61             robot.set_speed_profile(SpeedProfile.FAST)
62         else:
63             robot.set_speed_profile(SpeedProfile.DEFAULT)
64         robot.switch_mode(RobotMode.PLANNING_POS)
65         try:
66             robot.move_to_joint_pos([0, 0, 0, 0, 0, 0])
67             robot.move_to_joint_pos([0, -2, 2, 0, 1, 0])
68             robot.move_to_joint_pos([0, 0, 0, 0, 0, 0])
69         finally:
             robot.set_speed_profile(SpeedProfile.DEFAULT)

```

1.16 `mod` `airbot_examples.switch_mode`

Example of switching control mode.

Functions:

Name	Description
<code>switch_mode</code>	Switch control mode according to the given mode

1.16.1 `func` `airbot_examples.switch_mode.switch_mode()`

Switch control mode according to the given mode

Source code in `airbot_examples/switch_mode.py`



```

21 def switch_mode():
22     """
23     Switch control mode according to the given mode
24     """
25
26     import argparse
27
28     from airbot_py.arm import AIRBOTPlay, RobotMode
29
30     parser = argparse.ArgumentParser(description="Switch control mode example")
31     parser.add_argument(
32         "-m",
33         "--mode",
34         type=str,
35         choices=[
36             "planning_pos",
37             "planning_waypoints_pos",
38             "servo_joint_pos",
39             "servo_joint_vel",
40             "servo_cart_pose",
41             "servo_cart_twist",
42             "gravity_comp",
43         ],
44         default="planning_pos",
45         help="control mode to switch",
46     )
47     parser.add_argument(
48         "-u",
49         "--url",
50         type=str,
51         default="localhost",
52         help="Port to connect to the server",
53     )
54     parser.add_argument(
55         "-p", "--port", type=int, default=50051, help="Port to connect to the server"
56     )
57     args = parser.parse_args()
58
59     with AIRBOTPlay(url=args.url, port=args.port) as robot:
60         if args.mode == "planning_pos":
61             robot.switch_mode(RobotMode.PLANNING_POS)
62             print("Entered planning position mode")
63         elif args.mode == "planning_waypoints_pos":
64             robot.switch_mode(RobotMode.PLANNING_WAYPOINTS_POS)
65             print("Entered planning waypointss mode")
66         elif args.mode == "servo_joint_pos":
67             robot.switch_mode(RobotMode.SERVO_JOINT_POS)
68             print("Entered servo joint position mode")
69         elif args.mode == "servo_joint_vel":
70             robot.switch_mode(RobotMode.SERVO_JOINT_VEL)
71             print("Entered servo joint velocity mode.")
72         elif args.mode == "servo_cart_pose":
73             robot.switch_mode(RobotMode.SERVO_CART_POSE)
74             print("Entered servo cartesian pose mode.")
75         elif args.mode == "servo_cart_twist":
76             robot.switch_mode(RobotMode.SERVO_CART_TWIST)
77             print("Entered servo cartesian twist mode.")
78         elif args.mode == "gravity_comp":
79             robot.switch_mode(RobotMode.GRAVITY_COMP)
80             print("Entered gravity compensation mode.")
81             print(
82                 "\x1b[1;32mGravity compensation mode. You can drag the end of the robot arm
83             freely now.\x1b[0m"
84             )

```

```
else:
    print("Invalid mode")
```

84

85

1.17 `mod` `airbot_examples.task_follow`

Make one arm follow the movement of another arm.

Functions:

Name	Description
<code>follow</code>	Follow the lead robot arm

1.17.1 `func` `airbot_examples.task_follow.follow()`

Follow the lead robot arm

Source code in `airbot_examples/task_follow.py`



```

21 def follow():
22     """
23     Follow the lead robot arm
24     """
25     import argparse
26     import time
27
28     from airbot_py.arm import AIRBOTPlay, RobotMode, SpeedProfile
29
30     parser = argparse.ArgumentParser(description="Switch control mode example")
31     parser.add_argument(
32         "--lead-url",
33         type=str,
34         default="localhost",
35         help="URL to connect to the lead server",
36     )
37     parser.add_argument(
38         "--follow-url",
39         type=str,
40         default="localhost",
41         help="URL to connect to the follow server",
42     )
43     parser.add_argument(
44         "--lead-port", type=int, default=50051, help="Server port for arm to lead"
45     )
46     parser.add_argument(
47         "--follow-port", type=int, default=50052, help="Server port for arm to follow"
48     )
49
50     args = parser.parse_args()
51     if args.lead_port == args.follow_port and args.lead_url == args.follow_url:
52         raise ValueError(
53             "Lead and follow port, lead and follow url cannot be the same at the same time."
54         )
55
56     with AIRBOTPlay(
57         url=args.lead_url,
58         port=args.lead_port,
59     ) as robot1, AIRBOTPlay(
60         url=args.follow_url,
61         port=args.follow_port,
62     ) as robot2:
63         robot1.switch_mode(RobotMode.PLANNING_POS)
64         if (
65             sum(
66                 abs(i - j)
67                 for i, j in zip(robot1.get_joint_pos(), robot2.get_joint_pos())
68             )
69             > 0.1
70         ):
71             robot2.switch_mode(RobotMode.PLANNING_POS)
72             robot2.move_to_joint_pos(robot1.get_joint_pos())
73             time.sleep(1)
74         print("Lead robot arm is ready to follow.")
75         robot1.switch_mode(RobotMode.GRAVITY_COMP)
76         robot2.switch_mode(RobotMode.SERVO_JOINT_POS)
77         robot2.set_speed_profile(SpeedProfile.FAST)
78         try:
79             while True:
80                 robot2.servo_joint_pos(robot1.get_joint_pos())
81                 robot2.servo_eef_pos(robot1.get_eef_pos() or [])
82                 time.sleep(0.01)
83         finally:
84             robot1.switch_mode(RobotMode.PLANNING_POS)

```

```
robot2.switch_mode(RobotMode.PLANNING_POS)
robot2.set_speed_profile(SpeedProfile.DEFAULT)
```

```
84
85
86
```

1.18 `mod` `airbot_examples.task_kbd_ctrl`

Example application of controlling the robot arm using keyboard input.

Functions:

Name	Description
<code>task_kbd_ctrl</code>	Example application of controlling the robot arm using keyboard input.

1.18.1 `func` `airbot_examples.task_kbd_ctrl.task_kbd_ctrl()`

Example application of controlling the robot arm using keyboard input.

Source code in `airbot_examples/task_kbd_ctrl.py`



```

21 def task_kbd_ctrl():
22     """
23     Example application of controlling the robot arm using keyboard input.
24     """
25     import argparse
26     import curses
27     import math
28     import time
29
30     from airbot_py.arm import AIRBOTPlay, RobotMode, SpeedProfile
31
32     print(
33         "\x1b[1;33mThe robot arm is about to be move. Please make sure no obstacles exist in
34 the surrounding environment.\x1b[0m"
35     )
36     input("\x1b[1;35mPress Enter to continue...\x1b[0m")
37     print("\x1b[1;32mPress Ctrl+C to stop...\x1b[0m")
38
39     parser = argparse.ArgumentParser(description="Servo example")
40     parser.add_argument(
41         "-u",
42         "--url",
43         type=str,
44         default="localhost",
45         help="Port to connect to the server",
46     )
47     parser.add_argument(
48         "-p", "--port", type=int, default=50051, help="Port to connect to the server"
49     )
50
51     args = parser.parse_args()
52
53     def main(stdscr):
54         stdscr.nodelay(True)
55         state = 1
56         speed_profile = SpeedProfile.SLOW
57         with AIRBOTPlay(url=args.url, port=args.port) as robot:
58             robot.switch_mode(RobotMode.SERVO_CART_TWIST)
59             robot.set_speed_profile(speed_profile)
60             speed = 10
61             try:
62                 while True:
63                     key = stdscr.getch()
64                     if key != -1:
65                         if key == ord("\n"):
66                             if speed_profile == SpeedProfile.SLOW:
67                                 speed_profile = SpeedProfile.DEFAULT
68                             elif speed_profile == SpeedProfile.DEFAULT:
69                                 speed_profile = SpeedProfile.SLOW
70                             robot.set_speed_profile(speed_profile)
71                         if key == ord(" "):
72                             if state == 1:
73                                 robot.switch_mode(RobotMode.SERVO_JOINT_VEL)
74                                 state = 0
75                             elif state == 0:
76                                 robot.switch_mode(RobotMode.SERVO_CART_TWIST)
77                                 state = 1
78                         if state == 0:
79                             if key == ord("1"):
80                                 robot.servo_joint_vel([speed, 0.0, 0.0, 0.0, 0.0, 0.0])
81                             if key == ord("2"):
82                                 robot.servo_joint_vel([-speed, 0.0, 0.0, 0.0, 0.0, 0.0])
83                             if key == ord("3"):
84                                 robot.servo_joint_vel([0.0, speed, 0.0, 0.0, 0.0, 0.0])

```

```

if key == ord("4"):
    robot.servo_joint_vel([0.0, -speed, 0.0, 0.0, 0.0, 0.0])
if key == ord("5"):
    robot.servo_joint_vel([0.0, 0.0, speed, 0.0, 0.0, 0.0])
if key == ord("6"):
    robot.servo_joint_vel([0.0, 0.0, -speed, 0.0, 0.0, 0.0])
if key == ord("7"):
    robot.servo_joint_vel([0.0, 0.0, 0.0, speed, 0.0, 0.0])
if key == ord("8"):
    robot.servo_joint_vel([0.0, 0.0, 0.0, -speed, 0.0, 0.0])
if key == ord("9"):
    robot.servo_joint_vel([0.0, 0.0, 0.0, 0.0, speed, 0.0])
if key == ord("0"):
    robot.servo_joint_vel([0.0, 0.0, 0.0, 0.0, -speed, 0.0])
if key == ord("-"):
    robot.servo_joint_vel([0.0, 0.0, 0.0, 0.0, 0.0, speed])
if key == ord("="):
    robot.servo_joint_vel([0.0, 0.0, 0.0, 0.0, 0.0, -speed])
elif state == 1:
    if key == ord("a"):
        robot.servo_cart_twist(
            [[0.0, +speed, 0.0], [0.0, 0.0, 0.0]]
        )
    if key == ord("d"):
        robot.servo_cart_twist(
            [[0.0, -speed, 0.0], [0.0, 0.0, 0.0]]
        )
    if key == ord("w"):
        robot.servo_cart_twist(
            [[+speed, 0.0, 0.0], [0.0, 0.0, 0.0]]
        )
    if key == ord("s"):
        robot.servo_cart_twist(
            [[-speed, 0.0, 0.0], [0.0, 0.0, 0.0]]
        )
    if key == ord("q"):
        robot.servo_cart_twist(
            [[0.0, 0.0, +speed], [0.0, 0.0, 0.0]]
        )
    if key == ord("e"):
        robot.servo_cart_twist(
            [[0.0, 0.0, -speed], [0.0, 0.0, 0.0]]
        )
    if key == ord("o"):
        robot.servo_cart_twist(
            [[0.0, 0.0, 0.0], [+speed, 0.0, 0.0]]
        )
    if key == ord("u"):
        robot.servo_cart_twist(
            [[0.0, 0.0, 0.0], [-speed, 0.0, 0.0]]
        )
    if key == ord("i"):
        robot.servo_cart_twist(
            [[0.0, 0.0, 0.0], [0.0, +speed, 0.0]]
        )
    if key == ord("k"):
        robot.servo_cart_twist(
            [[0.0, 0.0, 0.0], [0.0, -speed, 0.0]]
        )
    if key == ord("j"):
        robot.servo_cart_twist(
            [[0.0, 0.0, 0.0], [0.0, 0.0, +speed]]
        )
    if key == ord("l"):
        robot.servo_cart_twist(

```

```

        [[0.0, 0.0, 0.0], [0.0, 0.0, -speed]]
    )
    if key == ord("\t"):
        robot.switch_mode(RobotMode.PLANNING_POS)
        robot.move_to_joint_pos(
            [
                0.0,
                -1.1840663267948965,
                1.1840663267948965,
                -90 / 360.0 * 2 * math.pi,
                90 / 360.0 * 2 * math.pi,
                0.0,
            ]
        )
        time.sleep(0.5)
        robot.switch_mode(RobotMode.SERVO_CART_TWIST)
        state = 1
    if key == ord("["):
        robot.servo_eef_pos([0.00])
    if key == ord("]"):
        robot.servo_eef_pos([0.07])
    if key == ord("z"): # 按 'q' 退出
        break
finally:
    robot.set_speed_profile(SpeedProfile.DEFAULT)

curses.wrapper(main)

```

84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120

```
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
```

1.19 `mod` `airbot_examples.task_swing`

Example application: making the robot arm swing in a sinusoidal motion in joint space.

Functions:

Name	Description
<code>task_swing</code>	Make the robot arm swing in a sinusoidal motion in joint space.

1.19.1 `func` `airbot_examples.task_swing.task_swing()`

Make the robot arm swing in a sinusoidal motion in joint space.

Source code in `airbot_examples/task_swing.py`




```

        / args.duration
    )
    joint_pos = [
        math.pi / 2 * timed_coef,
        -math.pi / 4 - math.pi / 4 * timed_coef,
        math.pi / 4 + math.pi / 4 * timed_coef,
        0,
        0,
        0,
    ]
    robot.serve_joint_pos(joint_pos)
    robot.serve_eef_pos([(1 - timed_coef) * 0.07 / 2])
    time.sleep(0.01)
finally:
    robot.set_speed_profile(SpeedProfile.DEFAULT)

```

```

84
85
86
87
88
89
90
91
92
93
94
95
96
97
98

```

1.20 `mod` `airbot_examples.task_wipe`

Example application: making the robot arm swing in a sinusoidal motion in cartesian space.

Functions:

Name	Description
<code>task_wipe</code>	Example application: making the robot arm swing in a sinusoidal motion in joint space.

1.20.1 `func` `airbot_examples.task_wipe.task_wipe()`

Example application: making the robot arm swing in a sinusoidal motion in joint space.

Source code in `airbot_examples/task_wipe.py`



```

21 def task_wipe():
22     """
23     Example application: making the robot arm swing in a sinusoidal motion in joint space.
24     """
25
26     import argparse
27     import json
28     import math
29     import time
30
31     print(
32         "\x1b[1;33mThe robot arm is about to be move. Please make sure no obstacles exist in
33 the surrounding environment.\x1b[0m"
34     )
35     input("\x1b[1;35mPress Enter to continue...\x1b[0m")
36     print("\x1b[1;32mPress Ctrl+C to stop...\x1b[0m")
37
38     from airbot_py.arm import AIRBOTPlay, RobotMode, SpeedProfile
39
40     parser = argparse.ArgumentParser(description="Servo example")
41     parser.add_argument(
42         "-u",
43         "--url",
44         type=str,
45         default="localhost",
46         help="Port to connect to the server",
47     )
48     parser.add_argument(
49         "-p", "--port", type=int, default=50051, help="Port to connect to the server"
50     )
51     parser.add_argument(
52         "-T",
53         "--duration",
54         type=float,
55         default=2.0,
56         help="Duration of the swing in seconds",
57     )
58     parser.add_argument(
59         "-S",
60         "--speed",
61         choices=["slow", "fast", "default"],
62         default="default",
63         help="Speed profile for the robot arm",
64     )
65     args = parser.parse_args()
66     with AIRBOTPlay(url=args.url, port=args.port) as robot:
67         if args.speed == "slow":
68             robot.set_speed_profile(SpeedProfile.SLOW)
69         elif args.speed == "fast":
70             robot.set_speed_profile(SpeedProfile.FAST)
71         else:
72             robot.set_speed_profile(SpeedProfile.DEFAULT)
73         robot.switch_mode(RobotMode.PLANNING_POS)
74         robot.move_to_cart_pose([[0.35, 0.0, 0.50], [0, 0, 0, 1]])
75         time.sleep(0.5)
76         robot.switch_mode(RobotMode.SERVO_JOINT_POS)
77
78         robot.switch_mode(RobotMode.SERVO_CART_POSE)
79         start = time.time_ns()
80         try:
81             while True:
82                 now = time.time_ns() - start - 0.5 * 1e9
83                 pose = [

```

```
        0.35,  
        0.15 * math.sin(now / 1e9 * 2 * math.pi / args.duration),  
        0.50,  
    ],  
    [0, 0, 0, 1],  
]  
robot.servo_cart_pose(pose)  
robot.servo_eef_pos([0])  
time.sleep(0.01)  
finally:  
    robot.set_speed_profile(SpeedProfile.DEFAULT)
```

```
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94
```