

## First Assignment for Experimental Robotics Laboratory

Generated by Doxygen 1.8.17



# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">person</a>	. . . . .	??
<a href="#">robot_control_server</a>	. . . . .	??
<a href="#">state_machine</a>	. . . . .	??



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

State	
state_machine.Normal	??
state_machine.Play	??
state_machine.Sleep	??



## Chapter 3

# Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">state_machine.Normal</a>		
Define <a href="#">Normal</a> state	. . . . .	??
<a href="#">state_machine.Play</a>		
Define <a href="#">Play</a> state	. . . . .	??
<a href="#">state_machine.Sleep</a>		
Define <a href="#">Sleep</a> state	. . . . .	??





## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

scripts/ <a href="#">person.py</a> . . . . .	??
scripts/ <a href="#">robot_control_server.py</a> . . . . .	??
scripts/ <a href="#">state_machine.py</a> . . . . .	??



## Chapter 5

# Namespace Documentation

### 5.1 person Namespace Reference

#### Functions

- def `person` ()  
*Publishes either a voice command or a location depending on the robot state.*

#### 5.1.1 Detailed Description

Implements two publishers. Mimics the behaviour of a person controlling the robot using voice commands or pointing gestures.

#### 5.1.2 Function Documentation

##### 5.1.2.1 `person()`

```
def person.person ( )
```

Publishes either a voice command or a location depending on the robot state.

### 5.2 robot\_control\_server Namespace Reference

#### Functions

- def `checkConsistency` (x, y)  
*Checks if the requested position is inside the map boundaries.*
- def `moveToDestination` (req)  
*Callback function for the service.*
- def `robotControlServer` ()  
*Client initialization.*

## Variables

- `timeScale` = `rospy.get_param("time_scale")`  
*The time scale of the simulation.*

### 5.2.1 Detailed Description

Implementation of a server/client pattern. Given a location, the module checks its consistency and then moves the robot accordingly.

### 5.2.2 Function Documentation

#### 5.2.2.1 `checkConsistency()`

```
def robot_control_server.checkConsistency (
    x,
    y )
```

Checks if the requested position is inside the map boundaries.

##### Parameters

<code>x</code>	The x position of the location
<code>y</code>	The y position of the location

##### Returns

The consistency of the location with respect to the map

#### 5.2.2.2 `moveToDestination()`

```
def robot_control_server.moveToDestination (
    req )
```

Callback function for the service.

##### Parameters

<code>req</code>	The client's requested location
------------------	---------------------------------

#### Returns

Whether the robot was able to reach the destination or not

#### 5.2.2.3 robotControlServer()

```
def robot_control_server.robotControlServer ( )
```

Client initialization.

### 5.2.3 Variable Documentation

#### 5.2.3.1 timeScale

```
robot_control_server.timeScale = rospy.get_param("time_scale")
```

The time scale of the simulation.

## 5.3 state\_machine Namespace Reference

### Classes

- class [Normal](#)  
*Define [Normal](#) state.*
- class [Play](#)  
*Define [Play](#) state.*
- class [Sleep](#)  
*Define [Sleep](#) state.*

### Functions

- def [robotControlCall](#) (x, y)  
*Calls the "robot\_control" service.*
- def [receivedVoiceCommand](#) (data)  
*Callback for the 'voice\_command' topic.*
- def [receivedPointingGesture](#) (data)  
*Callback for the 'pointing\_gesture' topic.*
- def [main](#) ()  
*State machine initialization.*

## Variables

- `personx` = `rospy.get_param("person/x")`  
*The person's position.*
- `persony` = `rospy.get_param("person/y")`
- `homex` = `rospy.get_param("home/x")`  
*The "home" position.*
- `homey` = `rospy.get_param("home/y")`
- `mapx` = `rospy.get_param("map/xmax")`  
*The map's boundaries.*
- `mapy` = `rospy.get_param("map/ymax")`
- `timeScale` = `rospy.get_param("time_scale")`  
*The time scale of the simulation.*
- `int sleepCounter` = 0  
*Counter.*
- `bool playState` = False  
*Flag for notifying the NORMAL state that the person published a play command.*

### 5.3.1 Detailed Description

Defines the different robot behaviours and the transitions between them. Available states are NORMAL, SLEEP and PLAY.

### 5.3.2 Function Documentation

#### 5.3.2.1 `main()`

```
def state_machine.main ( )
```

State machine initialization.

#### 5.3.2.2 `receivedPointingGesture()`

```
def state_machine.receivedPointingGesture (
    data )
```

Callback for the 'pointing\_gesture' topic.

##### Parameters

<code>data</code>	The pointed location
-------------------	----------------------

### 5.3.2.3 receivedVoiceCommand()

```
def state_machine.receivedVoiceCommand (
    data )
```

Callback for the 'voice\_command' topic.

#### Parameters

<i>data</i>	The voice command
-------------	-------------------

### 5.3.2.4 robotControlCall()

```
def state_machine.robotControlCall (
    x,
    y )
```

Calls the "robot\_control" service.

#### Parameters

<i>x</i>	The x position of the destination
<i>y</i>	The y position of the destination

## 5.3.3 Variable Documentation

### 5.3.3.1 homex

```
state_machine.homex = rospy.get_param("home/x")
```

The "home" position.

### 5.3.3.2 homey

```
state_machine.homey = rospy.get_param("home/y")
```

#### 5.3.3.3 mapx

```
state_machine.mapx = rospy.get_param("map/xmax")
```

The map's boundaries.

#### 5.3.3.4 mapy

```
state_machine.mapy = rospy.get_param("map/ymax")
```

#### 5.3.3.5 personx

```
state_machine.personx = rospy.get_param("person/x")
```

The person's position.

#### 5.3.3.6 persony

```
state_machine.persony = rospy.get_param("person/y")
```

#### 5.3.3.7 playState

```
bool state_machine.playState = False
```

Flag for notifying the NORMAL state that the person published a play command.

#### 5.3.3.8 sleepCounter

```
int state_machine.sleepCounter = 0
```

Counter.

#### 5.3.3.9 timeScale

```
state_machine.timeScale = rospy.get_param("time_scale")
```

The time scale of the simulation.



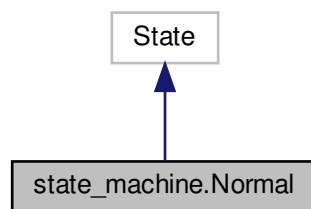
## Chapter 6

# Class Documentation

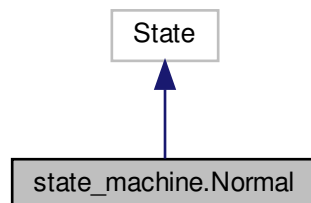
### 6.1 state\_machine.Normal Class Reference

Define [Normal](#) state.

Inheritance diagram for state\_machine.Normal:



Collaboration diagram for state\_machine.Normal:



## Public Member Functions

- `def \_\_init\_\_ (self)`
- `def execute (self, userdata)`

## Public Attributes

- [sleepThreshold](#)

### 6.1.1 Detailed Description

Define [Normal](#) state.

### 6.1.2 Constructor & Destructor Documentation

#### 6.1.2.1 `__init__()`

```
def state_machine.Normal.__init__ (  
    self )
```

### 6.1.3 Member Function Documentation

#### 6.1.3.1 `execute()`

```
def state_machine.Normal.execute (  
    self,  
    userdata )
```

### 6.1.4 Member Data Documentation

#### 6.1.4.1 `sleepThreshold`

```
state_machine.Normal.sleepThreshold
```

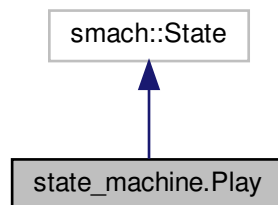
The documentation for this class was generated from the following file:

- `scripts/state\_machine.py`

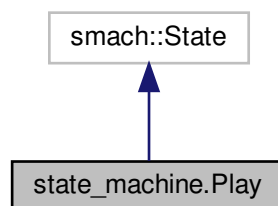
## 6.2 state\_machine.Play Class Reference

Define [Play](#) state.

Inheritance diagram for state\_machine.Play:



Collaboration diagram for state\_machine.Play:



### Public Member Functions

- def [\\_\\_init\\_\\_](#) (self)
- def [execute](#) (self, userdata)

### Public Attributes

- [timeThreshold](#)

### 6.2.1 Detailed Description

Define [Play](#) state.

## 6.2.2 Constructor & Destructor Documentation

### 6.2.2.1 `__init__()`

```
def state_machine.Play.__init__ (
    self )
```

## 6.2.3 Member Function Documentation

### 6.2.3.1 `execute()`

```
def state_machine.Play.execute (
    self,
    userdata )
```

## 6.2.4 Member Data Documentation

### 6.2.4.1 `timeThreshold`

```
state_machine.Play.timeThreshold
```

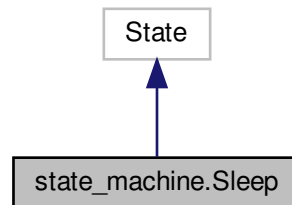
The documentation for this class was generated from the following file:

- [scripts/state\\_machine.py](#)

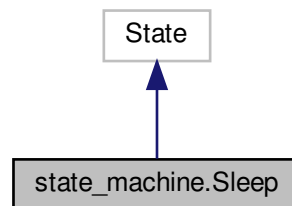
## 6.3 state\_machine.Sleep Class Reference

Define [Sleep](#) state.

Inheritance diagram for state\_machine.Sleep:



Collaboration diagram for state\_machine.Sleep:



### Public Member Functions

- `def \_\_init\_\_ (self)`
- `def execute (self, userdata)`

#### 6.3.1 Detailed Description

Define [Sleep](#) state.

#### 6.3.2 Constructor & Destructor Documentation

#### 6.3.2.1 `__init__()`

```
def state_machine.Sleep.__init__ (
    self )
```

### 6.3.3 Member Function Documentation

#### 6.3.3.1 `execute()`

```
def state_machine.Sleep.execute (
    self,
    userdata )
```

The documentation for this class was generated from the following file:

- [scripts/state\\_machine.py](#)

## Chapter 7

# File Documentation

### 7.1 CMakeLists.txt File Reference

### 7.2 scripts/person.py File Reference

#### Namespaces

- [person](#)

#### Functions

- `def person.person ()`  
*Publishes either a voice command or a location depending on the robot state.*

### 7.3 scripts/robot\_control\_server.py File Reference

#### Namespaces

- [robot\\_control\\_server](#)

#### Functions

- `def robot\_control\_server.checkConsistency (x, y)`  
*Checks if the requested position is inside the map boundaries.*
- `def robot\_control\_server.moveToDestination (req)`  
*Callback function for the service.*
- `def robot\_control\_server.robotControlServer ()`  
*Client initialization.*

## Variables

- `robot_control_server.timeScale` = `rospy.get_param("time_scale")`  
*The time scale of the simulation.*

## 7.4 scripts/state\_machine.py File Reference

### Classes

- class `state_machine.Normal`  
*Define *Normal* state.*
- class `state_machine.Sleep`  
*Define *Sleep* state.*
- class `state_machine.Play`  
*Define *Play* state.*

### Namespaces

- `state_machine`

### Functions

- def `state_machine.robotControlCall` (x, y)  
*Calls the "robot\_control" service.*
- def `state_machine.receivedVoiceCommand` (data)  
*Callback for the 'voice\_command' topic.*
- def `state_machine.receivedPointingGesture` (data)  
*Callback for the 'pointing\_gesture' topic.*
- def `state_machine.main` ()  
*State machine initialization.*

### Variables

- `state_machine.personx` = `rospy.get_param("person/x")`  
*The person's position.*
- `state_machine.persony` = `rospy.get_param("person/y")`
- `state_machine.homex` = `rospy.get_param("home/x")`  
*The "home" position.*
- `state_machine.homey` = `rospy.get_param("home/y")`
- `state_machine.mapx` = `rospy.get_param("map/xmax")`  
*The map's boundaries.*
- `state_machine.mapy` = `rospy.get_param("map/ymax")`
- `state_machine.timeScale` = `rospy.get_param("time_scale")`  
*The time scale of the simulation.*
- int `state_machine.sleepCounter` = 0  
*Counter.*
- bool `state_machine.playState` = False  
*Flag for notifying the NORMAL state that the person published a play command.*