

# Inverse Kinematics Solver for 6 Degrees of Freedom Robotic Manipulator

Generated by Doxygen 1.8.17



<b>1 Class Index</b>	<b>1</b>
1.1 Class List	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 Forward_Kinematics Class Reference	5
3.1.1 Detailed Description	5
3.1.2 Member Function Documentation	6
3.1.2.1 get_current_pose()	6
3.1.2.2 get_input_angles()	6
3.1.2.3 get_output_angles()	6
3.1.2.4 get_output_coordinates()	6
3.1.2.5 set_current_pose()	6
3.1.2.6 set_input_angles()	7
3.1.2.7 set_output_angles()	7
3.1.2.8 set_output_coordinates()	7
3.1.2.9 solve_FK()	9
3.2 Inverse_Kinematics Class Reference	9
3.2.1 Detailed Description	10
3.2.2 Member Function Documentation	10
3.2.2.1 convert_input_angles_to_rotation_matrix()	10
3.2.2.2 get_current_pose()	10
3.2.2.3 get_dh_a()	10
3.2.2.4 get_dh_alpha()	10
3.2.2.5 get_dh_d()	10
3.2.2.6 get_input_angles()	11
3.2.2.7 get_input_coordinates()	11
3.2.2.8 get_output_angles()	11
3.2.2.9 get_output_coordinates()	11
3.2.2.10 reset_pose()	11
3.2.2.11 set_current_pose()	11
3.2.2.12 set_dh_a()	11
3.2.2.13 set_dh_alpha()	12
3.2.2.14 set_dh_d()	12
3.2.2.15 set_input_angles()	12
3.2.2.16 set_input_coordinates()	12
3.2.2.17 set_output_angles()	12
3.2.2.18 set_output_coordinates()	12
3.2.2.19 solve_IK()	13
3.3 Manipulator Class Reference	13
3.3.1 Detailed Description	13

---

<b>4 File Documentation</b>	<b>15</b>
4.1 app/CMakeLists.txt File Reference	15
4.1.1 Function Documentation	15
4.1.1.1 add_executable()	15
4.1.1.2 target_link_libraries()	15
4.2 app/Forward_kinematics.cpp File Reference	16
4.2.1 Detailed Description	16
4.3 app/Inverse_kinematics.cpp File Reference	17
4.3.1 Detailed Description	17
4.4 app/main.cpp File Reference	18
4.4.1 Detailed Description	18
4.4.2 Macro Definition Documentation	19
4.4.2.1 PI	19
4.4.3 Function Documentation	19
4.4.3.1 main()	19
4.5 app/Manipulator.cpp File Reference	19
4.5.1 Detailed Description	20
4.6 include/Forward_kinematics.hpp File Reference	20
4.6.1 Detailed Description	21
4.6.2 Macro Definition Documentation	22
4.6.2.1 PI	22
4.7 include/Inverse_kinematics.hpp File Reference	22
4.7.1 Detailed Description	23
4.7.2 Macro Definition Documentation	23
4.7.2.1 PI	24
4.8 include/Manipulator.hpp File Reference	24
4.8.1 Detailed Description	24
<b>Index</b>	<b>27</b>

# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">matplotlibcpp::detail::_interpreter</a>	??
<a href="#">matplotlibcpp::detail::is_callable_impl&lt; true, T &gt;::Check&lt; U, U &gt;</a>	??
<a href="#">matplotlibcpp::detail::is_callable_impl&lt; true, T &gt;::Derived</a>	??
<a href="#">matplotlibcpp::detail::is_callable_impl&lt; true, T &gt;::Fallback</a>	??
<a href="#">Forward_Kinematics</a>	
The following Class contains all the methods,attributes of Forward Kinematics Class. It provide methods to solve the forward kinematics of a robotic manipulator	5
<a href="#">Inverse_Kinematics</a>	
The following Class contains all the methods,attributes of Inverse Kinematics Class. It provide methods to solve the inverse kinematics of a robotic manipulator	9
<a href="#">matplotlibcpp::detail::is_callable&lt; T &gt;</a>	??
<a href="#">matplotlibcpp::detail::is_callable_impl&lt; obj, T &gt;</a>	??
<a href="#">matplotlibcpp::detail::is_callable_impl&lt; false, T &gt;</a>	??
<a href="#">matplotlibcpp::detail::is_callable_impl&lt; true, T &gt;</a>	??
<a href="#">Manipulator</a>	
This Class will call the Forward and Inverse Kinematics	13
<a href="#">matplotlibcpp::Plot</a>	??
<a href="#">matplotlibcpp::detail::plot_impl&lt; IsYDataCallable &gt;</a>	??
<a href="#">matplotlibcpp::detail::plot_impl&lt; std::false_type &gt;</a>	??
<a href="#">matplotlibcpp::detail::plot_impl&lt; std::true_type &gt;</a>	??
<a href="#">matplotlibcpp::select_npy_type&lt; T &gt;</a>	??
<a href="#">matplotlibcpp::select_npy_type&lt; bool &gt;</a>	??
<a href="#">matplotlibcpp::select_npy_type&lt; double &gt;</a>	??
<a href="#">matplotlibcpp::select_npy_type&lt; float &gt;</a>	??
<a href="#">matplotlibcpp::select_npy_type&lt; int16_t &gt;</a>	??
<a href="#">matplotlibcpp::select_npy_type&lt; int32_t &gt;</a>	??
<a href="#">matplotlibcpp::select_npy_type&lt; int64_t &gt;</a>	??
<a href="#">matplotlibcpp::select_npy_type&lt; int8_t &gt;</a>	??
<a href="#">matplotlibcpp::select_npy_type&lt; uint16_t &gt;</a>	??
<a href="#">matplotlibcpp::select_npy_type&lt; uint32_t &gt;</a>	??
<a href="#">matplotlibcpp::select_npy_type&lt; uint64_t &gt;</a>	??
<a href="#">matplotlibcpp::select_npy_type&lt; uint8_t &gt;</a>	??



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

app/ <a href="#">Forward_kinematics.cpp</a>	
BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved . . . . .	16
app/ <a href="#">Inverse_kinematics.cpp</a>	
BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved . . . . .	17
app/ <a href="#">main.cpp</a>	
BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved . . . . .	18
app/ <a href="#">Manipulator.cpp</a>	
BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved . . . . .	19
include/ <a href="#">Forward_kinematics.hpp</a>	
BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved . . . . .	20
include/ <a href="#">Inverse_kinematics.hpp</a>	
BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved . . . . .	22
include/ <a href="#">Manipulator.hpp</a>	
BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved . . . . .	24
include/ <a href="#">matplotlibcpp.h</a> . . . . .	??





## Chapter 3

# Class Documentation

### 3.1 Forward\_Kinematics Class Reference

The following Class contains all the methods,attributes of Forward Kinematics Class. It provide methods to solve the forward kinematics of a robotic manipulator.

```
#include <Forward_kinematics.hpp>
```

#### Public Member Functions

- void [solve\\_FK](#) (const std::vector< double > &\_input\_joint\_angles)  
*this function will calculate the end effector position rom the given input\_joint\_angles.*
- void [set\\_output\\_coordinates](#) (const std::vector< double > &\_output\_joint\_coordinates)  
*It sets the output\_coordinates(input) to the output\_joint\_coordinates.*
- void [set\\_output\\_angles](#) (const std::vector< double > &\_output\_joint\_angles)  
*It sets the given input to output\_joint\_coordinates.*
- void [set\\_input\\_angles](#) (const std::vector< double > &\_input\_joint\_angles)  
*It sets the given input to input\_joint\_angles.*
- void [set\\_current\\_pose](#) (const std::vector< double > &\_current\_robot\_pose)  
*It sets the given input to current\_robot\_pose.*
- std::vector< double > [get\\_output\\_coordinates](#) ()  
*Getter method for returning output\_joint\_coordinates.*
- std::vector< double > [get\\_output\\_angles](#) ()  
*Getter Method for returning output\_joint\_angles.*
- std::vector< double > [get\\_current\\_pose](#) ()  
*Getter method for returning the current\_robot\_pose.*
- std::vector< double > [get\\_input\\_angles](#) ()  
*Getter method for getting the input\_joint\_angles.*

#### 3.1.1 Detailed Description

The following Class contains all the methods,attributes of Forward Kinematics Class. It provide methods to solve the forward kinematics of a robotic manipulator.

### 3.1.2 Member Function Documentation

#### 3.1.2.1 `get_current_pose()`

```
std::vector< double > Forward_Kinematics::get_current_pose ( )
```

Getter method for returning the `current_robot_pose`.

Returns

`current_robot_pose`

#### 3.1.2.2 `get_input_angles()`

```
std::vector< double > Forward_Kinematics::get_input_angles ( )
```

Getter method for getting the `input_joint_angles`.

Returns

`input_joint_angles`

#### 3.1.2.3 `get_output_angles()`

```
std::vector< double > Forward_Kinematics::get_output_angles ( )
```

Getter Method for returning `output_joint_angles`.

Returns

`output_joint_angles`

#### 3.1.2.4 `get_output_coordinates()`

```
std::vector< double > Forward_Kinematics::get_output_coordinates ( )
```

Getter method for returning `output_joint_coordinates`.

Returns

`output_joint_coordinates`

#### 3.1.2.5 `set_current_pose()`

```
void Forward_Kinematics::set_current_pose (
    const std::vector< double > & _current_robot_pose )
```

It sets the given input to `current_robot_pose`.

## Parameters

<code>_current_robot_pose</code>	
----------------------------------	--

## Returns

None

**3.1.2.6 set\_input\_angles()**

```
void Forward_Kinematics::set_input_angles (
    const std::vector< double > & _input_joint_angles )
```

It sets the given input to input\_joint\_angles.

## Parameters

<code>_input_joint_angles</code>	
----------------------------------	--

## Returns

None

**3.1.2.7 set\_output\_angles()**

```
void Forward_Kinematics::set_output_angles (
    const std::vector< double > & _output_joint_angles )
```

It sets the given input to output\_joint\_coordinates.

## Parameters

<code>_output_joint_angles</code>	
-----------------------------------	--

## Returns

None

**3.1.2.8 set\_output\_coordinates()**

```
void Forward_Kinematics::set_output_coordinates (
    const std::vector< double > & _output_joint_coordinates )
```

It sets the `output_coordinates(input)` to the `output_joint_coordinates`.

## Parameters

<code>_output_joint_coordinates</code>	
--	--

## Returns

None

**3.1.2.9 solve\_FK()**

```
void Forward_Kinematics::solve_FK (
    const std::vector< double > & _input_joint_angles )
```

this function will calculate the end effector position rom the given input\_joint\_angles.

## Parameters

<code>input_joint_angles</code>	these are the input joint angles of the robotic manipulator
---------------------------------	---

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

- [include/Forward\\_kinematics.hpp](#)
- [app/Forward\\_kinematics.cpp](#)

**3.2 Inverse\_Kinematics Class Reference**

The following Class contains all the methods,attributes of Inverse Kinematics Class. It provide methods to solve the inverse kinematics of a robotic manipulator.

```
#include <Inverse_kinematics.hpp>
```

**Public Member Functions**

- void [solve\\_IK](#) (const std::vector< double > &, const std::vector< double > &)
- void [set\\_input\\_coordinates](#) (const std::vector< double > &)
- void [set\\_output\\_coordinates](#) (const std::vector< double > &)
- void [set\\_output\\_angles](#) (const std::vector< double > &)
- void [set\\_input\\_angles](#) (const std::vector< double > &)
- void [set\\_current\\_pose](#) (const std::vector< double > &)
- void [set\\_dh\\_a](#) (const std::vector< double > &)
- void [set\\_dh\\_d](#) (const std::vector< double > &)
- void [set\\_dh\\_alpha](#) (const std::vector< double > &)
- std::vector< double > [get\\_input\\_coordinates](#) ()
- std::vector< double > [get\\_output\\_coordinates](#) ()
- std::vector< double > [get\\_input\\_angles](#) ()

- `std::vector< double > get_output_angles ()`
- `std::vector< double > get_current_pose ()`
- `std::vector< double > get_dh_a ()`
- `std::vector< double > get_dh_d ()`
- `std::vector< double > get_dh_alpha ()`
- `void reset_pose ()`
- `std::vector< double > convert_input_angles_to_rotation_matrix (const std::vector< double > &)`

### 3.2.1 Detailed Description

The following Class contains all the methods, attributes of Inverse Kinematics Class. It provides methods to solve the inverse kinematics of a robotic manipulator.

### 3.2.2 Member Function Documentation

#### 3.2.2.1 `convert_input_angles_to_rotation_matrix()`

```
std::vector< double > Inverse_Kinematics::convert_input_angles_to_rotation_matrix (
    const std::vector< double > & input_joint_angles )
```

#### 3.2.2.2 `get_current_pose()`

```
std::vector< double > Inverse_Kinematics::get_current_pose ( )
```

#### 3.2.2.3 `get_dh_a()`

```
std::vector< double > Inverse_Kinematics::get_dh_a ( )
```

#### 3.2.2.4 `get_dh_alpha()`

```
std::vector< double > Inverse_Kinematics::get_dh_alpha ( )
```

#### 3.2.2.5 `get_dh_d()`

```
std::vector< double > Inverse_Kinematics::get_dh_d ( )
```

### 3.2.2.6 get\_input\_angles()

```
std::vector< double > Inverse_Kinematics::get_input_angles ( )
```

### 3.2.2.7 get\_input\_coordinates()

```
std::vector< double > Inverse_Kinematics::get_input_coordinates ( )
```

### 3.2.2.8 get\_output\_angles()

```
std::vector< double > Inverse_Kinematics::get_output_angles ( )
```

### 3.2.2.9 get\_output\_coordinates()

```
std::vector< double > Inverse_Kinematics::get_output_coordinates ( )
```

### 3.2.2.10 reset\_pose()

```
void Inverse_Kinematics::reset_pose ( )
```

### 3.2.2.11 set\_current\_pose()

```
void Inverse_Kinematics::set_current_pose (
    const std::vector< double > & _current_robot_pose )
```

### 3.2.2.12 set\_dh\_a()

```
void Inverse_Kinematics::set_dh_a (
    const std::vector< double > & _dh_a )
```

### 3.2.2.13 set\_dh\_alpha()

```
void Inverse_Kinematics::set_dh_alpha (
    const std::vector< double > & _dh_alpha )
```

### 3.2.2.14 set\_dh\_d()

```
void Inverse_Kinematics::set_dh_d (
    const std::vector< double > & _dh_d )
```

### 3.2.2.15 set\_input\_angles()

```
void Inverse_Kinematics::set_input_angles (
    const std::vector< double > & _input_joint_angles )
```

### 3.2.2.16 set\_input\_coordinates()

```
void Inverse_Kinematics::set_input_coordinates (
    const std::vector< double > & _input_joint_coordinates )
```

### 3.2.2.17 set\_output\_angles()

```
void Inverse_Kinematics::set_output_angles (
    const std::vector< double > & _output_joint_angles )
```

### 3.2.2.18 set\_output\_coordinates()

```
void Inverse_Kinematics::set_output_coordinates (
    const std::vector< double > & _output_joint_coordinates )
```



#### 3.2.2.19 solve\_IK()

```
void Inverse_Kinematics::solve_IK (
    const std::vector< double > & input_joint_coordinates,
    const std::vector< double > & input_joint_angles )
```

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

- [include/Inverse\\_kinematics.hpp](#)
- [app/Inverse\\_kinematics.cpp](#)

## 3.3 Manipulator Class Reference

This Class will call the Forward and Inverse Kinematics.

```
#include <Manipulator.hpp>
```

### 3.3.1 Detailed Description

This Class will call the Forward and Inverse Kinematics.

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

- [include/Manipulator.hpp](#)



## Chapter 4

# File Documentation

### 4.1 app/CMakeLists.txt File Reference

#### Functions

- [add\\_executable](#) (shell-app main.cpp Manipulator.cpp Inverse\_kinematics.cpp Forward\_kinematics.cpp) find\_package(PythonLibs 2.7) target\_include\_directories(shell-app PRIVATE \$
- [target\\_link\\_libraries](#) (shell-app \${PYTHON\_LIBRARIES}) include\_directories(\$

#### 4.1.1 Function Documentation

##### 4.1.1.1 add\_executable()

```
add_executable (
    shell-app main.cpp Manipulator.cpp Inverse_kinematics.cpp Forward_kinematics.
    cpp )
```

##### 4.1.1.2 target\_link\_libraries()

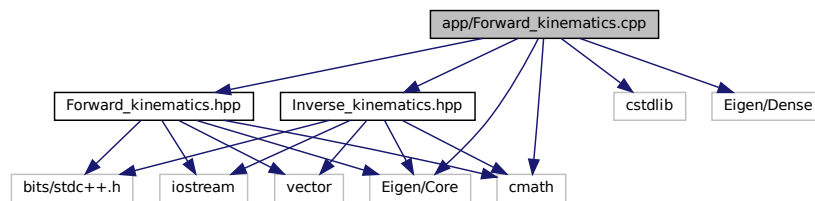
```
target_link_libraries (
    shell-app ${PYTHON_LIBRARIES} )
```

## 4.2 app/Forward\_kinematics.cpp File Reference

BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved.

```
#include "Forward_kinematics.hpp"
#include <cstdlib>
#include <cmath>
#include "Eigen/Core"
#include "Eigen/Dense"
#include "Inverse_kinematics.hpp"
```

Include dependency graph for Forward\_kinematics.cpp:



### 4.2.1 Detailed Description

BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#### Author

Rahul Karanam , Ameya Konkar

#### Copyright

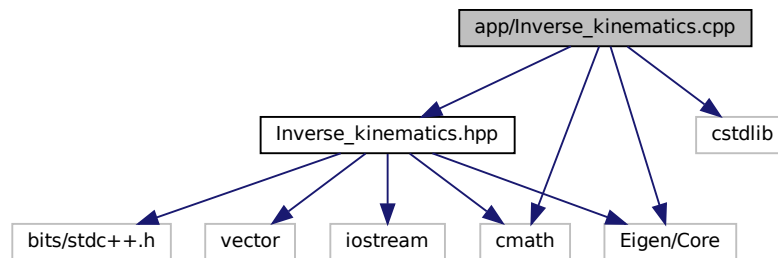
BSD 3-Clause License

This file contains the Forward Kinematics methods used to find out the end-effector coordinates of the robotic manipulator.

## 4.3 app/Inverse\_kinematics.cpp File Reference

BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved.

```
#include "Inverse_kinematics.hpp"
#include <cstdlib>
#include <cmath>
#include "Eigen/Core"
Include dependency graph for Inverse_kinematics.cpp:
```



### 4.3.1 Detailed Description

BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#### Author

Rahul Karanam , Ameya Konkar

#### Copyright

BSD 3-Clause License

This file contains the Forward Kinematics methods used to find out the end-effector coordinates of the robotic manipulator.

## 4.4 app/main.cpp File Reference

BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved.

```
#include "Eigen/Core"
#include "Eigen/Dense"
#include <iostream>
#include "Inverse_kinematics.hpp"
#include "Forward_kinematics.hpp"
#include "matplotlibcpp.h"
#include <cmath>
```

Include dependency graph for main.cpp:



### Macros

- #define `PI` 3.14

### Functions

- int `main` ()

We use this main function to output the output joint coordinates for the given input\_coordinates.

#### 4.4.1 Detailed Description

BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

**Author**

Rahul Karanam , Ameya Konkar

**Copyright**

BSD 3-Clause License

This is our main source code file. It calls inverse Kinematics to implement our IK solver to simulate our path.

## 4.4.2 Macro Definition Documentation

### 4.4.2.1 PI

```
#define PI 3.14
```

## 4.4.3 Function Documentation

### 4.4.3.1 main()

```
int main ( )
```

We use this main function to output the output joint coordinates for the given input\_coordinates.

**Returns**

0;

## 4.5 app/Manipulator.cpp File Reference

BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved.

### 4.5.1 Detailed Description

BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#### Author

Rahul Karanam , Ameya Konkar

#### Copyright

BSD 3-Clause License

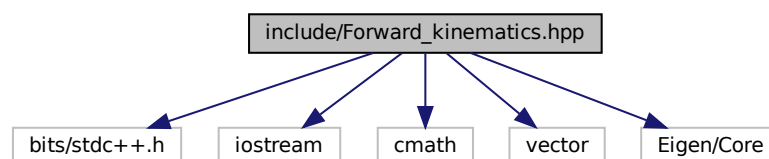
This file contains the Forward Kinematics methods used to find out the end-effector coordinates of the robotic manipulator.

## 4.6 include/Forward\_kinematics.hpp File Reference

BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved.

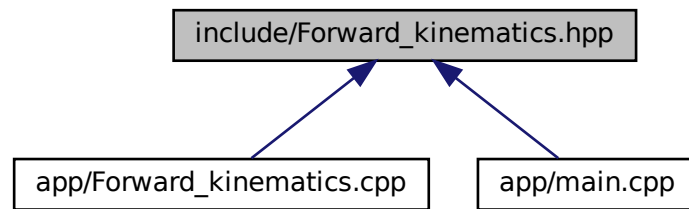
```
#include <bits/stdc++.h>
#include <iostream>
#include <cmath>
#include <vector>
#include "Eigen/Core"
```

Include dependency graph for Forward\_kinematics.hpp:





This graph shows which files directly or indirectly include this file:



## Classes

- class [Forward\\_Kinematics](#)

*The following Class contains all the methods, attributes of Forward Kinematics Class. It provide methods to solve the forward kinematics of a robotic manipulator.*

## Macros

- `#define` [PI](#) 3.14

### 4.6.1 Detailed Description

BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

**Author**

Rahul Karanam , Ameya Konkar

**Copyright**

BSD 3-Clause License

This header file contains the Forward Kinematics class members and attributes Class to call solve\_FK,getter and setter methods

## 4.6.2 Macro Definition Documentation

### 4.6.2.1 PI

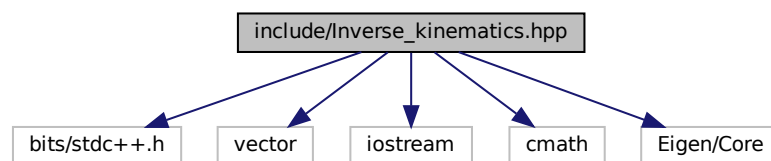
```
#define PI 3.14
```

## 4.7 include/Inverse\_kinematics.hpp File Reference

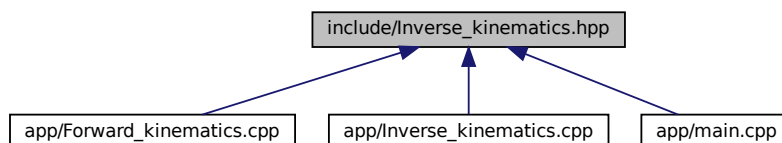
BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved.

```
#include <bits/stdc++.h>
#include <vector>
#include <iostream>
#include <cmath>
#include "Eigen/Core"
```

Include dependency graph for Inverse\_kinematics.hpp:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Inverse\\_Kinematics](#)

*The following Class contains all the methods, attributes of Inverse Kinematics Class. It provide methods to solve the inverse kinematics of a robotic manipulator.*

## Macros

- #define [PI](#) 3.14

### 4.7.1 Detailed Description

BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#### Author

Rahul Karanam , Ameya Konkar

#### Copyright

BSD 3-Clause License

This header file contains the Forward Kinematics class members and attributes Class to call solve\_FK,getter and setter methods

### 4.7.2 Macro Definition Documentation

#### 4.7.2.1 PI

```
#define PI 3.14
```

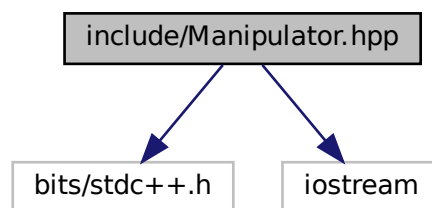
## 4.8 include/Manipulator.hpp File Reference

BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved.

```
#include <bits/stdc++.h>
```

```
#include <iostream>
```

Include dependency graph for Manipulator.hpp:



### Classes

- class [Manipulator](#)

*This Class will call the Forward and Inverse Kinematics.*

#### 4.8.1 Detailed Description

BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

**Author**

Rahul Karanam , Ameya Konkar

**Copyright**

BSD 3-Clause License

This Class will call the Forward Kinematics and Inverse Kinematics.



# Index

- add\_executable
  - CMakeLists.txt, [15](#)
- app/CMakeLists.txt, [15](#)
- app/Forward\_kinematics.cpp, [16](#)
- app/Inverse\_kinematics.cpp, [17](#)
- app/main.cpp, [18](#)
- app/Manipulator.cpp, [19](#)
- CMakeLists.txt
  - add\_executable, [15](#)
  - target\_link\_libraries, [15](#)
- convert\_input\_angles\_to\_rotation\_matrix
  - Inverse\_Kinematics, [10](#)
- Forward\_Kinematics, [5](#)
  - get\_current\_pose, [6](#)
  - get\_input\_angles, [6](#)
  - get\_output\_angles, [6](#)
  - get\_output\_coordinates, [6](#)
  - set\_current\_pose, [6](#)
  - set\_input\_angles, [7](#)
  - set\_output\_angles, [7](#)
  - set\_output\_coordinates, [7](#)
  - solve\_FK, [9](#)
- Forward\_kinematics.hpp
  - PI, [22](#)
- get\_current\_pose
  - Forward\_Kinematics, [6](#)
  - Inverse\_Kinematics, [10](#)
- get\_dh\_a
  - Inverse\_Kinematics, [10](#)
- get\_dh\_alpha
  - Inverse\_Kinematics, [10](#)
- get\_dh\_d
  - Inverse\_Kinematics, [10](#)
- get\_input\_angles
  - Forward\_Kinematics, [6](#)
  - Inverse\_Kinematics, [10](#)
- get\_input\_coordinates
  - Inverse\_Kinematics, [11](#)
- get\_output\_angles
  - Forward\_Kinematics, [6](#)
  - Inverse\_Kinematics, [11](#)
- get\_output\_coordinates
  - Forward\_Kinematics, [6](#)
  - Inverse\_Kinematics, [11](#)
- include/Forward\_kinematics.hpp, [20](#)
- include/Inverse\_kinematics.hpp, [22](#)
- include/Manipulator.hpp, [24](#)
- Inverse\_Kinematics, [9](#)
  - convert\_input\_angles\_to\_rotation\_matrix, [10](#)
  - get\_current\_pose, [10](#)
  - get\_dh\_a, [10](#)
  - get\_dh\_alpha, [10](#)
  - get\_dh\_d, [10](#)
  - get\_input\_angles, [10](#)
  - get\_input\_coordinates, [11](#)
  - get\_output\_angles, [11](#)
  - get\_output\_coordinates, [11](#)
  - reset\_pose, [11](#)
  - set\_current\_pose, [11](#)
  - set\_dh\_a, [11](#)
  - set\_dh\_alpha, [11](#)
  - set\_dh\_d, [12](#)
  - set\_input\_angles, [12](#)
  - set\_input\_coordinates, [12](#)
  - set\_output\_angles, [12](#)
  - set\_output\_coordinates, [12](#)
  - solve\_IK, [12](#)
- Inverse\_kinematics.hpp
  - PI, [23](#)
- main
  - main.cpp, [19](#)
- main.cpp
  - main, [19](#)
  - PI, [19](#)
- Manipulator, [13](#)
- PI
  - Forward\_kinematics.hpp, [22](#)
  - Inverse\_kinematics.hpp, [23](#)
  - main.cpp, [19](#)
- reset\_pose
  - Inverse\_Kinematics, [11](#)
- set\_current\_pose
  - Forward\_Kinematics, [6](#)
  - Inverse\_Kinematics, [11](#)
- set\_dh\_a
  - Inverse\_Kinematics, [11](#)
- set\_dh\_alpha
  - Inverse\_Kinematics, [11](#)
- set\_dh\_d
  - Inverse\_Kinematics, [12](#)
- set\_input\_angles
  - Forward\_Kinematics, [7](#)

- Inverse\_Kinematics, [12](#)
- set\_input\_coordinates
  - Inverse\_Kinematics, [12](#)
- set\_output\_angles
  - Forward\_Kinematics, [7](#)
  - Inverse\_Kinematics, [12](#)
- set\_output\_coordinates
  - Forward\_Kinematics, [7](#)
  - Inverse\_Kinematics, [12](#)
- solve\_FK
  - Forward\_Kinematics, [9](#)
- solve\_IK
  - Inverse\_Kinematics, [12](#)
- target\_link\_libraries
  - CMakeLists.txt, [15](#)