

Inverse Kinematics Solver for 6 Degree of Freedom Robotic Manipulator

Generated by Doxygen 1.8.17

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
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

3.3.2 Member Function Documentation	14
3.3.2.1 print_IK_solver()	14
3.3.3 Member Data Documentation	14
3.3.3.1 F	14
3.3.3.2 I	14
4 File Documentation	15
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	25
4.8.2 Macro Definition Documentation	25
4.8.2.1 PI	25
Index	27

Chapter 1

Class Index

1.1 Class List

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

[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 5

[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 9

[Manipulator](#)

This Class will call the Forward and Inverse Kinematics 13

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

app/ Forward_kinematics.cpp	
BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved	16
app/ Inverse_kinematics.cpp	
BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved	17
app/ main.cpp	
BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved	18
app/ Manipulator.cpp	
BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved	19
include/ Forward_kinematics.hpp	
BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved	20
include/ Inverse_kinematics.hpp	
BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved	22
include/ Manipulator.hpp	
BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved	24

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:

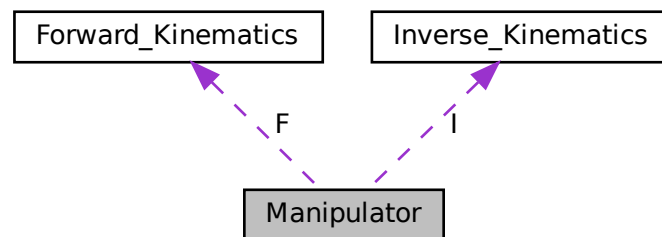
- [include/Inverse_kinematics.hpp](#)
- [app/Inverse_kinematics.cpp](#)

3.3 Manipulator Class Reference

This Class will call the Forward and Inverse Kinematics.

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

Collaboration diagram for Manipulator:



Public Member Functions

- void [print_IK_solver](#) ()
It will print out the IK and FK for the given inputs.

Public Attributes

- [Inverse_Kinematics](#) I
- [Forward_Kinematics](#) F

3.3.1 Detailed Description

This Class will call the Forward and Inverse Kinematics.

3.3.2 Member Function Documentation

3.3.2.1 `print_IK_solver()`

```
void Manipulator::print_IK_solver ( )
```

It will print out the IK and FK for the given inputs.

Returns

None

3.3.3 Member Data Documentation

3.3.3.1 `F`

```
Forward\_Kinematics Manipulator::F
```

3.3.3.2 `I`

```
Inverse\_Kinematics Manipulator::I
```

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

- [include/Manipulator.hpp](#)
- [app/Manipulator.cpp](#)

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()

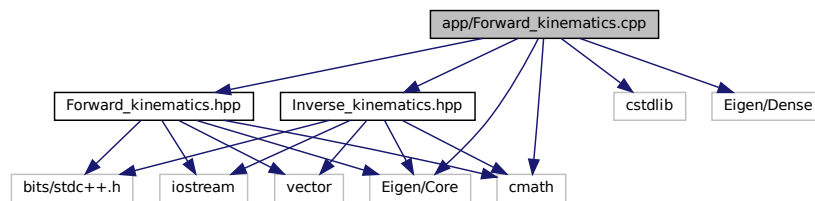
```
target_link_libraries (
    shell-app ${PYTHON_LIBRARIES} )
```

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

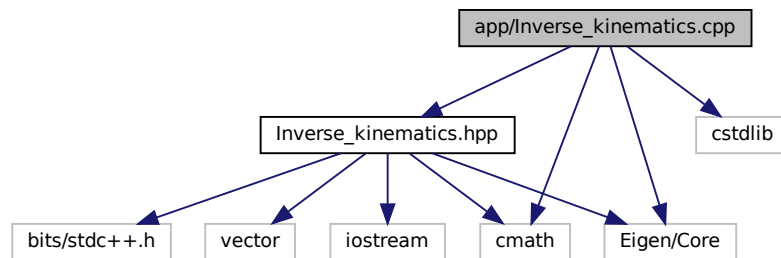
BSD 3-Clause License

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.

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

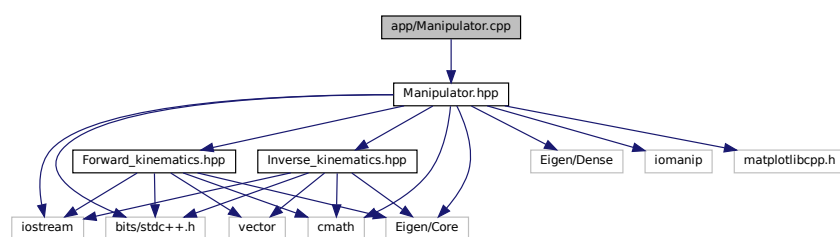
```
0;
```

4.5 app/Manipulator.cpp File Reference

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

```
#include "Manipulator.hpp"
```

Include dependency graph for Manipulator.cpp:



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

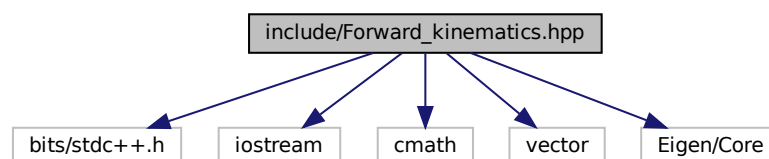
This file contains the Forward Kinematics methods used to find out the end-effector coordinates of the robotic manipulator.

4.6 include/Forward_kinematics.hpp File Reference

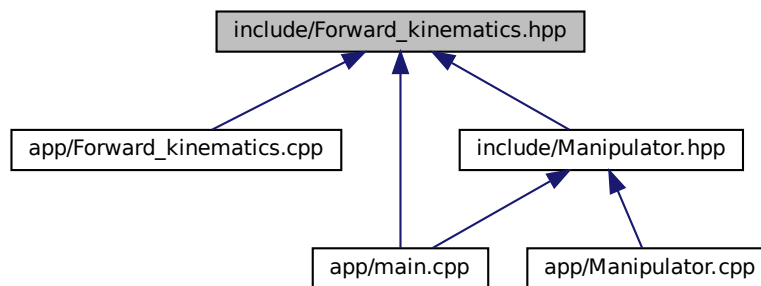
BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved.

```
#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**

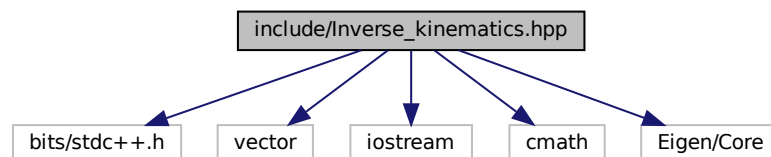
```
#define PI 3.14
```

4.7 include/Inverse_kinematics.hpp File Reference

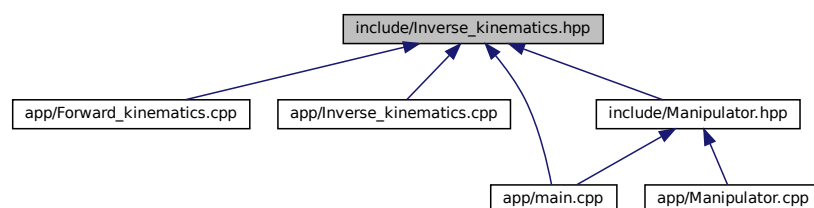
BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved.

```
#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

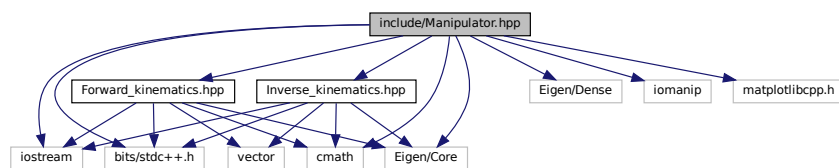
```
#define PI 3.14
```

4.8 include/Manipulator.hpp File Reference

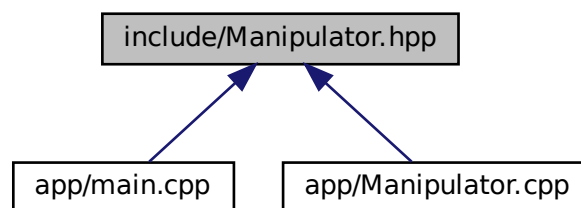
BSD 3-Clause License Copyright (c) 2021, ACME Robotics, Rahul Karanam , Ameya Konkar All rights reserved.

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

Include dependency graph for Manipulator.hpp:



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



Classes

- class [Manipulator](#)

This Class will call the Forward and Inverse Kinematics.

Macros

- #define `PI` 3.14

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.

4.8.2 Macro Definition Documentation

4.8.2.1 `PI`

```
#define PI 3.14
```


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

F

 Manipulator, 14
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

I

 Manipulator, 14
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
 F, 14
 I, 14
 print_IK_solver, 14
Manipulator.hpp
 PI, 25

PI

 Forward_kinematics.hpp, 22
 Inverse_kinematics.hpp, 23
 main.cpp, 19
 Manipulator.hpp, 25
print_IK_solver
 Manipulator, 14

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](#)