

IK SOLVER

PHASE 1

Generated by Doxygen 1.8.17

1 6 DOF Manipulator - Inverse Kinematics Solver	1
1.1 Project Description	1
1.1.1 Overview	1
1.1.2 Method	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Class Documentation	9
5.1 ForwardKinematics Class Reference	9
5.1.1 Detailed Description	10
5.1.2 Member Function Documentation	10
5.1.2.1 calculate_TF()	10
5.1.2.2 solve_fk()	10
5.2 InverseKinematics Class Reference	11
5.2.1 Detailed Description	12
5.2.2 Constructor & Destructor Documentation	12
5.2.2.1 InverseKinematics()	12
5.2.3 Member Function Documentation	12
5.2.3.1 get_eff_angles()	12
5.2.3.2 get_eff_position()	12
5.2.3.3 set_eff_angles()	12
5.2.3.4 set_eff_position()	13
5.2.3.5 solve_ik()	13
5.3 RobotParameters Class Reference	13
5.3.1 Detailed Description	14
5.3.2 Constructor & Destructor Documentation	14
5.3.2.1 RobotParameters()	14
5.3.3 Member Function Documentation	14
5.3.3.1 get_dh_parameters()	14
5.3.3.2 get_robot_angles()	15
5.3.3.3 set_robot_angles()	15
5.3.4 Member Data Documentation	15
5.3.4.1 robot_name	15
6 File Documentation	17
6.1 app/CMakeLists.txt File Reference	17
6.1.1 Function Documentation	17

6.1.1.1 add_executable()	17
6.2 test/CMakeLists.txt File Reference	17
6.2.1 Function Documentation	17
6.2.1.1 set()	18
6.3 app/forward_kinematics.cpp File Reference	18
6.3.1 Detailed Description	18
6.4 app/inverse_kinematics.cpp File Reference	19
6.5 app/main.cpp File Reference	19
6.5.1 Detailed Description	20
6.5.2 Function Documentation	20
6.5.2.1 main()	20
6.6 test/main.cpp File Reference	21
6.6.1 Detailed Description	21
6.6.2 Function Documentation	21
6.6.2.1 main()	22
6.7 app/robot_parameters.cpp File Reference	22
6.7.1 Detailed Description	22
6.8 docs/introduction.txt File Reference	23
6.9 include/forward_kinematics.hpp File Reference	23
6.9.1 Detailed Description	23
6.10 include/inverse_kinematics.hpp File Reference	24
6.10.1 Detailed Description	25
6.11 include/robot_parameters.hpp File Reference	26
6.11.1 Detailed Description	26
6.12 test/code_test.cpp File Reference	27
6.12.1 Detailed Description	28
6.12.2 Function Documentation	28
6.12.2.1 TEST() [1/8]	28
6.12.2.2 TEST() [2/8]	28
6.12.2.3 TEST() [3/8]	29
6.12.2.4 TEST() [4/8]	29
6.12.2.5 TEST() [5/8]	29
6.12.2.6 TEST() [6/8]	29
6.12.2.7 TEST() [7/8]	29
6.12.2.8 TEST() [8/8]	29

Chapter 1

6 DOF Manipulator - Inverse Kinematics Solver

1.1 Project Description

1.1.1 Overview

The aim of this project is to design and develop an inverse kinematics solver for a 6-axis Manipulator at ACME Robotics. This implementation is based on the Kuka-KR 5 Manipulator, however it can provide a solution for any 6 axis manipulator with a spherical wrist configuration. Our software will compute and simulate a trajectory based on the path coordinates provided and will be integrated as a module into a future project over at ACME Robotics.

1.1.2 Method

Our implementation includes three methods, two for solving inverse and forwards kinematics for the manipulator, and one method to store the attributes of the specified robot arm. The solution for inverse kinematics will be calculated and stored as joint angles for each path coordinate provided. The forward kinematics solver will calculate end effector position based on the joint values and robot parameters provided. Denavit Hartenberg representation will be employed for both the inverse and forward kinematics solvers. A two sprint Agile Iterative Process approach and test driven development style is utilized in the making of this project.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

RobotParameters	13
ForwardKinematics	9
InverseKinematics	11

Chapter 3

Class Index

3.1 Class List

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

ForwardKinematics	
Definition of the Forward Kinematics Class	9
InverseKinematics	
Definition of Inverse Kinematics Class	11
RobotParameters	
Definition of the Robot Parameter Class	13

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

app/forward_kinematics.cpp	
Program to define the Methods of Forward Kinematics Class	18
app/inverse_kinematics.cpp	19
app/main.cpp	
Program to execute the inverse kinematics and forward kinematics	19
app/robot_parameters.cpp	
Program to define the Methods of Robot Parameters Class	22
include/forward_kinematics.hpp	
Definition of Forward Kinematics class and Declaration of its Methods	23
include/inverse_kinematics.hpp	24
include/robot_parameters.hpp	
Definition of Robot Parameters class and Declaration of its Methods	26
test/code_test.cpp	
Program to perform unit testing	27
test/main.cpp	
Program to use google test for unit testing	21

Chapter 5

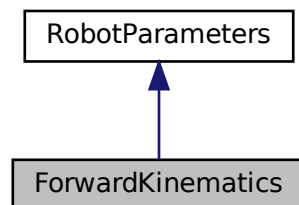
Class Documentation

5.1 ForwardKinematics Class Reference

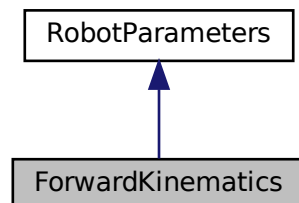
Definition of the Forward Kinematics Class.

```
#include <forward_kinematics.hpp>
```

Inheritance diagram for ForwardKinematics:



Collaboration diagram for ForwardKinematics:



Public Member Functions

- `Eigen::Matrix< double, 4, 4 > calculate_TF (int i)`
Calculate the DH transformation matrix for each joint pair.
- `Eigen::Matrix< double, 4, 4 > solve_fk ()`
Solve the forward kinematics for manipulator.

Additional Inherited Members

5.1.1 Detailed Description

Definition of the Forward Kinematics Class.

5.1.2 Member Function Documentation

5.1.2.1 `calculate_TF()`

```
Matrix< double, 4, 4 > ForwardKinematics::calculate_TF (
    int i )
```

Calculate the DH transformation matrix for each joint pair.

Parameters

<i>i</i>	integer value denoting the row of <code>_dh_matrix</code> to be considered
----------	--

Returns

`Eigen::Matrix<double, 4, 4>` Returns the transformation matrix

5.1.2.2 `solve_fk()`

```
Matrix< double, 4, 4 > ForwardKinematics::solve_fk ( )
```

Solve the forward kinematics for manipulator.

Returns

`Eigen::Matrix<double, 4, 4>` Returns the final Homogeneous transformation matrix

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

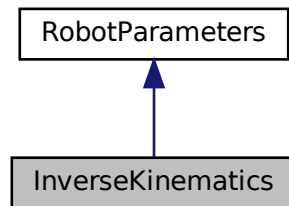
- `include/forward_kinematics.hpp`
- `app/forward_kinematics.cpp`

5.2 InverseKinematics Class Reference

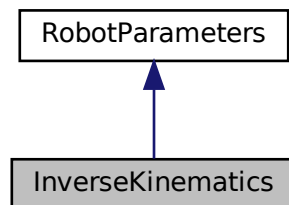
Definition of Inverse Kinematics Class.

```
#include <inverse_kinematics.hpp>
```

Inheritance diagram for InverseKinematics:



Collaboration diagram for InverseKinematics:



Public Member Functions

- [InverseKinematics](#) ()
Construct a new Inverse Kinematics object to assign default values.
- `std::vector< double > get_eff_angles ()`
Gets the end effector angles.
- `std::vector< double > get_eff_position ()`
Gets the end effector position.
- `void set_eff_angles (std::vector< double > eff_angles)`
Sets the end effector angles.
- `void set_eff_position (std::vector< double > eff_position)`
Sets the end effector position.
- `std::vector< double > solve_ik ()`
Method to solve the inverse kinematics of the Manipulator.

Additional Inherited Members

5.2.1 Detailed Description

Definition of Inverse Kinematics Class.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 InverseKinematics()

```
InverseKinematics::InverseKinematics ( )
```

Construct a new Inverse Kinematics object to assign default values.

5.2.3 Member Function Documentation

5.2.3.1 get_eff_angles()

```
vector< double > InverseKinematics::get_eff_angles ( )
```

Gets the end effector angles.

Returns

std::vector<double> Returns a vector containing end effector angles

5.2.3.2 get_eff_position()

```
vector< double > InverseKinematics::get_eff_position ( )
```

Gets the end effector position.

Returns

std::vector<double> Returns a vector containing end effector position

5.2.3.3 set_eff_angles()

```
void InverseKinematics::set_eff_angles (
    std::vector< double > eff_angles )
```

Sets the end effector angles.

Parameters

<i>eff_angles</i>	Vector containing end effector angles
-------------------	---------------------------------------

5.2.3.4 set_eff_position()

```
void InverseKinematics::set_eff_position (
    std::vector< double > eff_position )
```

Sets the end effector position.

Parameters

<i>eff_position</i>	Vector containing end effector position
---------------------	---

5.2.3.5 solve_ik()

```
vector< double > InverseKinematics::solve_ik ( )
```

Method to solve the inverse kinematics of the Manipulator.

Returns

`std::vector<double>`

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

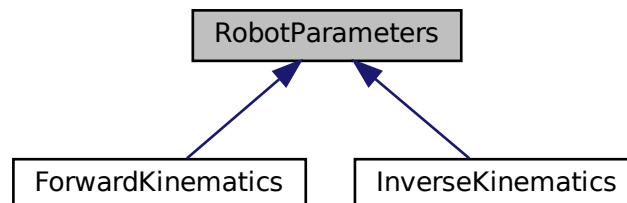
- [include/inverse_kinematics.hpp](#)
- [app/inverse_kinematics.cpp](#)

5.3 RobotParameters Class Reference

Definition of the Robot Parameter Class.

```
#include <robot_parameters.hpp>
```

Inheritance diagram for RobotParameters:



Public Member Functions

- [RobotParameters](#) ()
Construct a new Robot Parameters object to assign default values.
- Eigen::MatrixXd [get_dh_parameters](#) ()
Compute the dh parameters matrix.
- std::vector< double > [get_robot_angles](#) ()
Gets the robot angles.
- void [set_robot_angles](#) (std::vector< double > robot_angles)
Sets the robot angles.

Public Attributes

- std::string [robot_name](#)

5.3.1 Detailed Description

Definition of the Robot Parameter Class.

5.3.2 Constructor & Destructor Documentation

5.3.2.1 RobotParameters()

```
RobotParameters::RobotParameters ( )
```

Construct a new Robot Parameters object to assign default values.

5.3.3 Member Function Documentation

5.3.3.1 get_dh_parameters()

```
MatrixXd RobotParameters::get_dh_parameters ( )
```

Compute the dh parameters matrix.

Returns

Eigen::MatrixXd Returns DH matrix

5.3.3.2 get_robot_angles()

```
vector< double > RobotParameters::get_robot_angles ( )
```

Gets the robot angles.

Returns

std::vector<double> Returns the robot angles

5.3.3.3 set_robot_angles()

```
void RobotParameters::set_robot_angles (
    std::vector< double > robot_angles )
```

Sets the robot angles.

Parameters

<i>robot_angles</i>	Sets the robot angles from the ik solver output
---------------------	---

5.3.4 Member Data Documentation

5.3.4.1 robot_name

```
std::string RobotParameters::robot_name
```

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

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

Chapter 6

File Documentation

6.1 app/CMakeLists.txt File Reference

Functions

- [add_executable](#) (ik_solver main.cpp robot_parameters.cpp forward_kinematics.cpp) include_directories(\$

6.1.1 Function Documentation

6.1.1.1 add_executable()

```
add_executable (
    ik_solver main.cpp robot_parameters.cpp forward_kinematics.  cpp )
```

6.2 test/CMakeLists.txt File Reference

Functions

- [set](#) (GTEST_SHUFFLE 1) [add_executable](#)(code_test main.cpp code_test.cpp ../app/robot_parameters.cpp ../app/forward_kinematics.cpp ../app/inverse_kinematics.cpp) target_include_directories(code_test PUBLIC ../vendor/googletest/googletest/include \$

6.2.1 Function Documentation

6.2.1.1 set()

```
set (
    GTEST_SHUFFLE 1 )
```

6.3 app/forward_kinematics.cpp File Reference

Program to define the Methods of Forward Kinematics Class.

```
#include "../include/forward_kinematics.hpp"
#include <iostream>
Include dependency graph for forward_kinematics.cpp:
```

6.3.1 Detailed Description

Program to define the Methods of Forward Kinematics Class.

Author

Driver : Tanmay Haldankar (tanmayh@umd.edu), Navigator: Sanchit Kedia (sanchit@terpmail.umd.edu), Design Keeper: Qamar Syed (qsyed@umd.edu)

Version

0.12

Date

2022-10-18

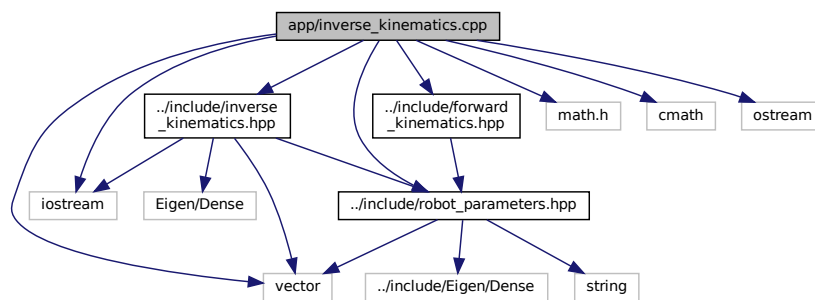
Copyright

MIT License (c)

6.4 app/inverse_kinematics.cpp File Reference

```
#include "../include/inverse_kinematics.hpp"
#include <math.h>
#include <cmath>
#include <iostream>
#include <ostream>
#include <vector>
#include "../include/forward_kinematics.hpp"
#include "../include/robot_parameters.hpp"
```

Include dependency graph for inverse_kinematics.cpp:

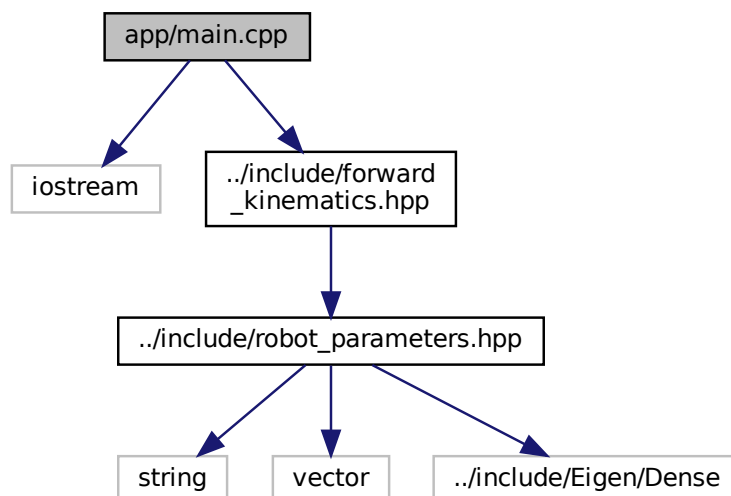


6.5 app/main.cpp File Reference

Program to execute the inverse kinematics and forward kinematics.

```
#include <iostream>
#include "../include/forward_kinematics.hpp"
```

Include dependency graph for main.cpp:



Functions

- `int main ()`
Main Function.

6.5.1 Detailed Description

Program to execute the inverse kinematics and forward kinematics.

Author

Driver : Sanchit Kedia (sanchit@terpmail.umd.edu), Navigator: Tanmay Haldankar (tanmayh@umd.edu), Design Keeper: Qamar Syed (qsyed@umd.edu)

Version

0.2

Date

2022-10-18

Copyright

MIT License (c)

6.5.2 Function Documentation

6.5.2.1 main()

```
int main ( )
```

Main Function.

Returns

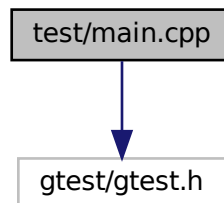
int 0

6.6 test/main.cpp File Reference

Program to use google test for unit testing.

```
#include <gtest/gtest.h>
```

Include dependency graph for main.cpp:



Functions

- int `main` (int argc, char **argv)

6.6.1 Detailed Description

Program to use google test for unit testing.

Author

Driver : Sanchit Kedia (sanchit@terpmail.umd.edu), Navigator: Tanmay Haldankar (tanmayh@umd.edu), Design Keeper: Qamar Syed (qsyed@umd.edu)

Version

0.1

Date

2022-10-13

Copyright

MIT License (c)

6.6.2 Function Documentation

6.6.2.1 main()

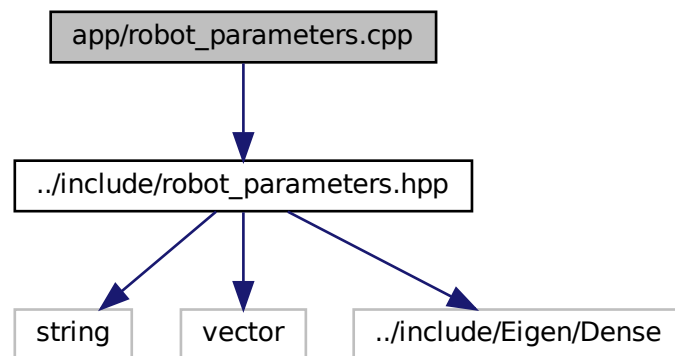
```
int main (
    int argc,
    char ** argv )
```

6.7 app/robot_parameters.cpp File Reference

Program to define the Methods of Robot Parameters Class.

```
#include "../include/robot_parameters.hpp"
```

Include dependency graph for robot_parameters.cpp:



6.7.1 Detailed Description

Program to define the Methods of Robot Parameters Class.

Author

Driver : Sanchit Kedia (sanchit@terpmail.umd.edu), Navigator: Tanmay Haldankar (tanmayh@umd.edu), Design Keeper: Qamar Syed (qsyed@umd.edu)

Version

0.11

Date

2022-10-13

Copyright

MIT License (c)

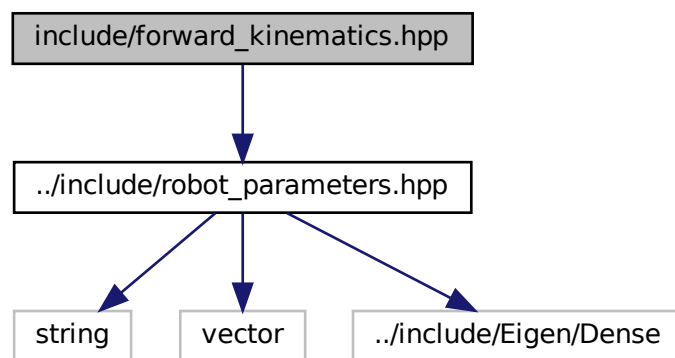
6.8 docs/introduction.txt File Reference

6.9 include/forward_kinematics.hpp File Reference

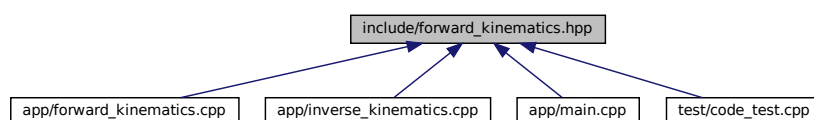
Definition of Forward Kinematics class and Declaration of its Methods.

```
#include "../include/robot_parameters.hpp"
```

Include dependency graph for forward_kinematics.hpp:



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



Classes

- class [ForwardKinematics](#)
Definition of the Forward Kinematics Class.

6.9.1 Detailed Description

Definition of Forward Kinematics class and Declaration of its Methods.

Author

Driver : Tanmay Haldankar (tanmayh@umd.edu), Navigator: Sanchit Kedia (sanchit@terpmail.umd.edu), Design Keeper: Qamar Syed (qsyed@umd.edu)

Version

0.1

Date

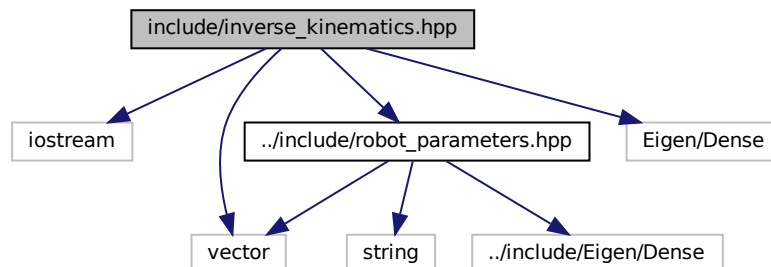
2022-10-18

Copyright

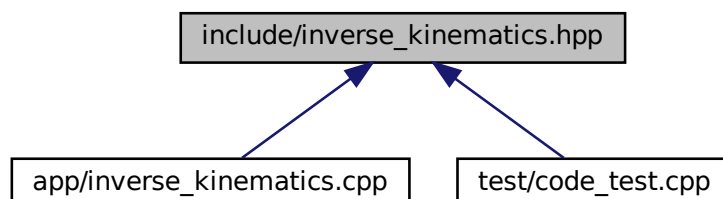
MIT License (c)

6.10 include/inverse_kinematics.hpp File Reference

```
#include <iostream>
#include <vector>
#include "../include/robot_parameters.hpp"
#include "Eigen/Dense"
Include dependency graph for inverse_kinematics.hpp:
```



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



Classes

- class [InverseKinematics](#)

Definition of Inverse Kinematics Class.

6.10.1 Detailed Description

Author

Driver : Sanchit Kedia (sanchit@terpmail.umd.edu), Navigator: Tanmay Haldankar (tanmayh@umd.edu), Design Keeper: Qamar Syed (qsyed@umd.edu)

Version

0.1

Date

2022-10-21

Copyright

MIT License (c)

Author

Driver : Tanmay Haldankar (tanmayh@umd.edu), Navigator: Sanchit Kedia (sanchit@terpmail.umd.edu), Design Keeper: Qamar Syed (qsyed@umd.edu)

Version

0.1

Date

2022-10-21

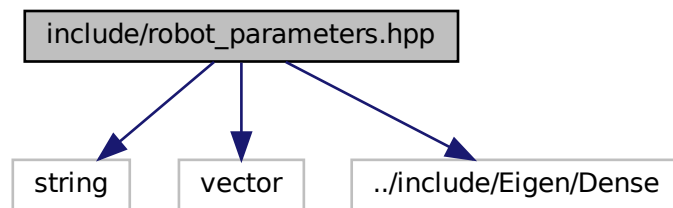
Copyright

MIT License (c)

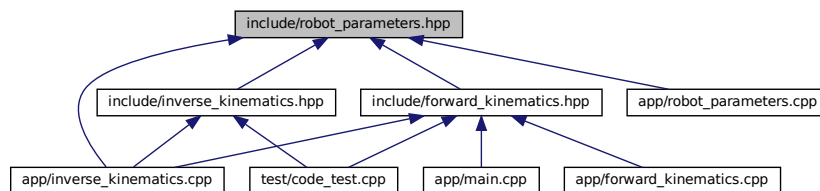
6.11 include/robot_parameters.hpp File Reference

Definition of Robot Parameters class and Declaration of its Methods.

```
#include <string>
#include <vector>
#include "../include/Eigen/Dense"
Include dependency graph for robot_parameters.hpp:
```



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



Classes

- class [RobotParameters](#)

Definition of the Robot Parameter Class.

6.11.1 Detailed Description

Definition of Robot Parameters class and Declaration of its Methods.

Author

Driver : Sanchit Kedia (sanchit@terpmail.umd.edu), Navigator: Tanmay Haldankar (tanmayh@umd.edu), Design Keeper: Qamar Syed (qsyed@umd.edu)

Version

0.1

Date

2022-10-15

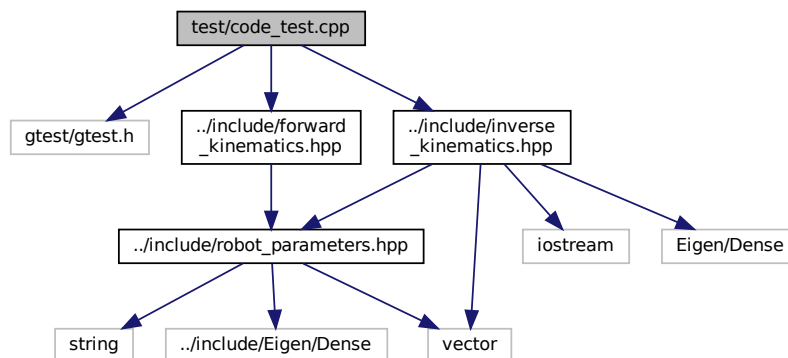
Copyright

MIT License (c)

6.12 test/code_test.cpp File Reference

Program to perform unit testing.

```
#include <gtest/gtest.h>
#include "../include/forward_kinematics.hpp"
#include "../include/inverse_kinematics.hpp"
Include dependency graph for code_test.cpp:
```



Functions

- **TEST** (Robot_Parameters, CheckAngles)
Construct a new TEST to check if the robot angles vector is empty.
- **TEST** (Robot_Parameters, CheckDH)
Construct a new TEST to check the size of the DH Parameters matrix.
- **TEST** (Robot_Parameters, CheckSetAngles)
Construct a new TEST to check if the robot angles are being set correctly.
- **TEST** (Forward_Kinematics, check_calculateTF)
Construct a new TEST to check the size of the DH transformation matrix.
- **TEST** (Forward_Kinematics, check_solvefk)
Construct a new TEST to check the size of final homogeneous transformation matrix.
- **TEST** (Inverse_Kinematics, Check_solveik)
- **TEST** (Inverse_Kinematics, CheckSetIKAngles)
- **TEST** (Inverse_Kinematics, CheckSetIKPosition)

6.12.1 Detailed Description

Program to perform unit testing.

Author

Driver : Sanchit Kedia (sanchit@terpmail.umd.edu), Navigator: Tanmay Haldankar (tanmayh@umd.edu), Design Keeper: Qamar Syed (qsyed@umd.edu)

Version

0.3

Date

2022-10-15

Copyright

MIT License (c)

6.12.2 Function Documentation

6.12.2.1 TEST() [1/8]

```
TEST (
    Forward_Kinematics ,
    check_calculateTF )
```

Construct a new TEST to check the size of the DH transformation matrix.

6.12.2.2 TEST() [2/8]

```
TEST (
    Forward_Kinematics ,
    check_solvefk )
```

Construct a new TEST to check the size of final homogeneous transformation matrix.

6.12.2.3 TEST() [3/8]

```
TEST (
    Inverse_Kinematics ,
    Check_solveik )
```

6.12.2.4 TEST() [4/8]

```
TEST (
    Inverse_Kinematics ,
    CheckSetIKAngles )
```

6.12.2.5 TEST() [5/8]

```
TEST (
    Inverse_Kinematics ,
    CheckSetIKPosition )
```

6.12.2.6 TEST() [6/8]

```
TEST (
    Robot_Parameters ,
    CheckAngles )
```

Construct a new TEST to check if the robot angles vector is empty.

6.12.2.7 TEST() [7/8]

```
TEST (
    Robot_Parameters ,
    CheckDH )
```

Construct a new TEST to check the size of the DH Parameters matrix.

6.12.2.8 TEST() [8/8]

```
TEST (
    Robot_Parameters ,
    CheckSetAngles )
```

Construct a new TEST to check if the robot angles are being set correctly.

Index

- add_executable
 - CMakeLists.txt, [17](#)
- app/CMakeLists.txt, [17](#)
- app/forward_kinematics.cpp, [18](#)
- app/inverse_kinematics.cpp, [19](#)
- app/main.cpp, [19](#)
- app/robot_parameters.cpp, [22](#)
- calculate_TF
 - ForwardKinematics, [10](#)
- CMakeLists.txt
 - add_executable, [17](#)
 - set, [17](#)
- code_test.cpp
 - TEST, [28, 29](#)
- docs/introduction.txt, [23](#)
- ForwardKinematics, [9](#)
 - calculate_TF, [10](#)
 - solve_fk, [10](#)
- get_dh_parameters
 - RobotParameters, [14](#)
- get_eff_angles
 - InverseKinematics, [12](#)
- get_eff_position
 - InverseKinematics, [12](#)
- get_robot_angles
 - RobotParameters, [14](#)
- include/forward_kinematics.hpp, [23](#)
- include/inverse_kinematics.hpp, [24](#)
- include/robot_parameters.hpp, [26](#)
- InverseKinematics, [11](#)
 - get_eff_angles, [12](#)
 - get_eff_position, [12](#)
 - InverseKinematics, [12](#)
 - set_eff_angles, [12](#)
 - set_eff_position, [13](#)
 - solve_ik, [13](#)
- main
 - main.cpp, [20, 21](#)
- main.cpp
 - main, [20, 21](#)
- robot_name
 - RobotParameters, [15](#)
- RobotParameters, [13](#)
 - get_dh_parameters, [14](#)
 - get_robot_angles, [14](#)
 - robot_name, [15](#)
 - RobotParameters, [14](#)
 - set_robot_angles, [15](#)
- set
 - CMakeLists.txt, [17](#)
- set_eff_angles
 - InverseKinematics, [12](#)
- set_eff_position
 - InverseKinematics, [13](#)
- set_robot_angles
 - RobotParameters, [15](#)
- solve_fk
 - ForwardKinematics, [10](#)
- solve_ik
 - InverseKinematics, [13](#)
- TEST
 - code_test.cpp, [28, 29](#)
- test/CMakeLists.txt, [17](#)
- test/code_test.cpp, [27](#)
- test/main.cpp, [21](#)