MotionDesk

Calculating and Streaming Motion Data

For MotionDesk Blockset 2.6.3

Release 2021-A – May 2021

How to Contact dSPACE

Mail:	dSPACE GmbH Rathenaustraße 26 33102 Paderborn Germany
Tel.:	+49 5251 1638-0
Fax:	+49 5251 16198-0
E-mail:	info@dspace.de
Web:	http://www.dspace.com

How to Contact dSPACE Support

If you encounter a problem when using dSPACE products, contact your local dSPACE representative:

- Local dSPACE companies and distributors: http://www.dspace.com/go/locations
- For countries not listed, contact dSPACE GmbH in Paderborn, Germany.
  Tel.: +49 5251 1638-941 or e-mail: support@dspace.de

You can also use the support request form: http://www.dspace.com/go/supportrequest. If you are logged on to mydSPACE, you are automatically identified and do not need to add your contact details manually.

If possible, always provide the relevant dSPACE License ID or the serial number of the CmContainer in your support request.

Software Updates and Patches

dSPACE strongly recommends that you download and install the most recent patches for your current dSPACE installation. Visit http://www.dspace.com/go/patches for software updates and patches.

Important Notice

Contents

Reference Information 49

About This Document

Contents

This document introduces you to the calculation of motion data in a simulation and streaming of the data to the MotionDesk PC. It provides the information required for modeling in MATLAB/Simulink and handcoding with C functions.

Target group

Engineers that want to stream motion data to the MotionDesk PC must read this document.

Required knowledge

Basic knowledge in handling the dSPACE software and MotionDesk for scene creation and animation is assumed.

Knowledge of hardware-in-the-loop (HIL) real-time simulation, offline simulation using VEOS and implementing models in MATLAB/Simulink is advised.

Symbols

dSPACE user documentation uses the following symbols:

Symbol	Description
⚠ DANGER	Indicates a hazardous situation that, if not avoided, will result in death or serious injury.
⚠ WARNING	Indicates a hazardous situation that, if not avoided, could result in death or serious injury.
⚠ CAUTION	Indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.
NOTICE	Indicates a hazard that, if not avoided, could result in property damage.
Note	Indicates important information that you should take into account to avoid malfunctions.
Tip	Indicates tips that can make your work easier.

Symbol	Description
	Indicates a link that refers to a definition in the glossary, which you can find at the end of the document unless stated otherwise.
	Precedes the document title in a link that refers to another document.

Naming conventions

dSPACE user documentation uses the following naming conventions:

%name% Names enclosed in percent signs refer to environment variables for file and path names.

< > Angle brackets contain wildcard characters or placeholders for variable file and path names, etc.

Special folders

Common Program Data folder A standard folder for application-specific configuration data that is used by all users.

%PROGRAMDATA%\dSPACE\<InstallationGUID>\<ProductName>

or

%PROGRAMDATA%\dSPACE\<ProductName>\<VersionNumber>

Documents folder A standard folder for user-specific documents.

%USERPROFILE%\Documents\dSPACE\<ProductName>\
<VersionNumber>

Local Program Data folder A standard folder for application-specific configuration data that is used by the current, non-roaming user.

%USERPROFILE%\AppData\Local\dSPACE\<InstallationGUID>\
<ProductName>

Accessing dSPACE Help and PDF Files

After you install and decrypt dSPACE software, the documentation for the installed products is available in dSPACE Help and as PDF files.

dSPACE Help (local) You can open your local installation of dSPACE Help:

- On its home page via Windows Start Menu
- On specific content using context-sensitive help via **F1**

dSPACE Help (Web) You can access the Web version of dSPACE Help at www.dspace.com/go/help.

To access the Web version, you must have a *mydSPACE* account.

PDF files You can access PDF files via the icon in dSPACE Help. The PDF opens on the first page.

Basics and Instructions

Where to go from here

Information in this section

Basics of Calculating and Streaming Motion Data

Introduction

Provides general information on the MotionDesk Blockset for the calculation and streaming of motion data and how to prepare the simulation model.

Where to go from here

Information in this section

Motion Data Calculation and Streaming Models

Introduction

To connect the simulation software with MotionDesk, you must build a Simulink model to calculate and stream the motion data.

Where to go from here

Information in this section

Methods for Adapting Motion Data Models

Introduction

dSPACE provides the MotionDesk Blockset to build Simulink models to calculate and stream the motion data between the simulation and MotionDesk. A MotionDesk Service Library of C functions is also available to adapt handcoded models for the streaming of the motion data to MotionDesk.

MotionDesk Blockset

You can use the MotionDesk Blockset to connect the simulation with MotionDesk for MotionDesk simulations. The blockset streams the calculated motion data to MotionDesk. The model can be built in MATLAB Simulink.

Note

This blockset must not be used for sensor simulation with camera, fish-eye, laser, lidar and radar sensors.

MotionDesk Service Library

You can use the MotionDesk service library to adapt handcoded models in C programming language.

You must integrate MotionDesk service library C functions into the handcoded models. The library includes all the C functions you need to data calculate and stream the motion data between the ASM real-time application and MotionDesk.

Note

- You cannot integrate the control of a MotionDesk Multi-PC Interface Kit into a handcoded model.
- You cannot use the MotionDesk Service Library to handcode models for use with SCALEXIO systems or with VEOS. You must use the MotionDesk Blockset if you work with these systems.
- You cannot use the MotionDesk Service Library to handcode models for connection with sensor simulation applications.
- You must use the Model and Sensor Interface Blockset for Sensor Simulation.

Related topics

Basics

Working with the MotionDesk Blockset 19
Working with the MotionDesk Service Library 46

Blockset Supported Platforms

Introduction

The MotionDesk Blockset is supported on specific dSPACE simulation platforms.

MotionDesk Blockset supported platforms

MotionDesk and the MotionDesk Blockset support the following platforms:

- Simulink
- SCALEXIO Systems
- MicroAutoBox III
- VEOS and VEOS on Linux

RTI platforms

- DS1006
- DS1007
- MicroLabBox (DS1202)
- MicroAutoBox II (DS1401)

Note

To work with Sensor Simulation, you must use the Model and Sensor Interface Blockset. For more information, refer to Basics of the Model and Sensor Interface Blockset (Model and Sensor Interface Blockset Manual 📖).

Related topics

References

Simulation Data Mathematical Principles

Introduction

To prepare simulation models, you must know the mathematical principles for the movement and transformation of objects used by the simulation data for visualization in MotionDesk and for Sensor Simulation.

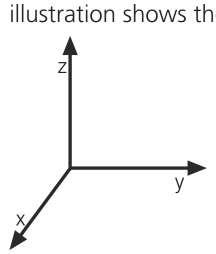
Where to go from here

Information in this section

Coordinate System Used in MotionDesk

Introduction

MotionDesk works with a right-hand Cartesian coordinate system. The following illustration shows the coordinate system as used in MotionDesk:



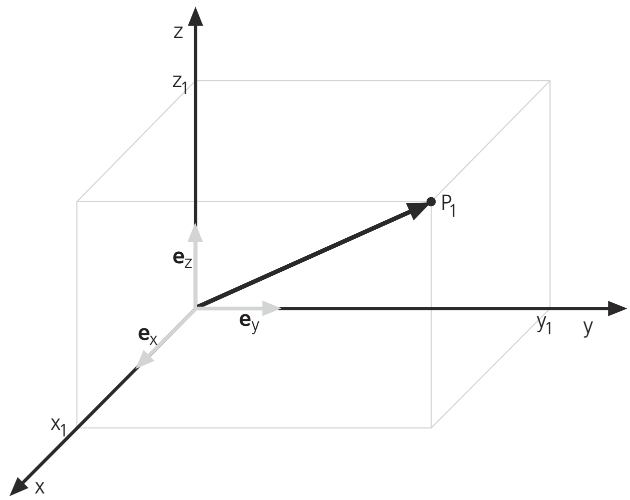
Vertex in a 3-D Space

Introduction

A position of a point (vertex) can be described using a translation vector.

Translation vector

To describe the position of a point (vertex) in a right-hand Cartesian coordinate system, three mutually orthogonal unit vectors ($\vec{e}_x$, $\vec{e}_y$, $\vec{e}_z$) are used, refer to the following illustration.



A point in space is referenced by a vector:

$$P = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} = x_1\vec{e}_x + y_1\vec{e}_y + z_1\vec{e}_z$$

The vector describes the distance between the origin and a point in the coordinate system. In MotionDesk, this vector is called the translation vector.

Homogeneous Transformation

Introduction

Adding a fourth dimension to the 3-D vector simplifies the calculation of 3-D operations, such as rotation, translation, and scaling. The transformations can therefore be calculated by multiplication with a 4 x 4 matrix.

Translation

The following matrix moves an object in the x-, y-, z- directions:

$$T = \begin{pmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Scaling

The following matrix scales an object in the x-, y-, z- directions:

$$S = \begin{pmatrix} x & 0 & 0 & 0 \\ 0 & y & 0 & 0 \\ 0 & 0 & z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation

In a three-dimensional coordinate system, three different rotations are possible. The different rotation matrices are shown below.

x-axis The following matrix rotates an object around its x-axis:

$$R(x,\alpha) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

y-axis The following matrix rotates an object around its y-axis:

$$R(y,\alpha) = \begin{pmatrix} \cos(\alpha) & 0 & \sin(\alpha) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

z-axis The following matrix rotates an object around its z-axis:

$$R(z,\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The equations above are for rotation around only one axis. If an object is rotated around two or three axes, the rotations must be done one after the other. The most common descriptions for rotation based on three independent variables are Euler angles (mechanics, gyroscopics) and roll, pitch and yaw angles (nautics and aeronautics).

Related topics

Basics

Cardan Roll, Pitch, and Yaw Angles.......... 16
Euler Angles.......... 15

Euler Angles

Introduction

The Euler angles are the three angles of rotation of an object around a fixed axis.

Euler angles

Using the Euler angles, a 3-D object is rotated around its axes in the following manner:

1. Rotation around the z-axis at angle ψ
2. Rotation around the y-axis at angle θ
3. Rotation around the actual z-axis at angle ϕ

The rotations are expressed in the following equation:

$$R_{\text{Euler}} = R_{\text{Euler}}(\phi, \theta, \psi) = R_{(z,\phi)}\, R_{(y,\theta)}\, R_{(z,\psi)}$$

The rotation matrix can be calculated by the following equation.

$$R_{\text{Euler}} =$$

$$\begin{pmatrix} \cos\phi\cos\theta\cos\psi - \sin\phi\sin\psi & -\cos\phi\cos\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\sin\psi \\ \sin\phi\cos\theta\cos\psi + \cos\phi\sin\psi & -\sin\phi\cos\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\sin\psi \\ -\sin\theta\cos\psi & \sin\theta\sin\psi & \cos\theta \end{pmatrix}$$

Related topics

Basics

Cardan Roll, Pitch, and Yaw Angles

Introduction

The Cardan angles of roll, pitch, and yaw angles are three angles that describe the orientation of a body in a three-dimensional space.

Cardan roll, pitch, and yaw angles (z-y'-x'' convention)

Using the Cardan angles of roll, pitch and yaw, a 3-D object is rotated around its axes in the following manner if the z-y'-x'' convention is used:

1. Rotation around the z-axis at angle ϕ
2. Rotation around the y-axis at angle θ
3. Rotation around the actual x-axis at angle ψ

The rotations are expressed in the following equation:

$$R_{\text{RPY}} = R_{\text{RPY}}(\phi, \theta, \psi) = R_{(z,\phi)}\, R_{(y,\theta)}\, R_{(x,\psi)}$$

The rotation matrix can be calculated with the following equation.

$R_{\mathrm{RPY}} =$

$$\begin{pmatrix} \cos\phi\,\cos\theta & \cos\phi\,\sin\theta\,\sin\psi\ -\sin\phi\,\cos\psi & \cos\phi\,\sin\theta\,\cos\psi\ +\sin\phi\,\sin\psi \\ \sin\phi\,\cos\theta & -\sin\phi\,\sin\theta\,\sin\psi\ +\cos\phi\,\cos\psi & \sin\phi\,\sin\psi\,\cos\psi\ -\cos\phi\,\sin\psi \\ -\sin\theta & \cos\theta\,\sin\psi & \cos\theta\,\cos\psi \end{pmatrix}$$

Cardan roll, pitch, and yaw angles (z-x'-y'' convention)

Using the Cardan angles of roll, pitch, and yaw, a 3-D object is rotated around its axes in the following manner if the z-x'-y'' convention is used:

1. Rotation around the z-axis at angle ϕ
2. Rotation around the x-axis at angle ψ
3. Rotation around the actual y-axis at angle θ

The rotations are expressed in the following equation:

$$R_{\mathrm{RPY}} = R_{\mathrm{RPY}}(\phi, \psi, \theta) = R_{(z,\phi)}\, R_{(x,\psi)}\, R_{(y,\theta)}$$

The rotation matrix can be calculated with the following equation.

$$R_{\mathrm{RYP}} = \begin{pmatrix} \cos\phi\cos\psi - \sin\phi\sin\theta\sin\psi & -\sin\phi\cos\theta & \cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi \\ \sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \cos\phi\cos\theta & \sin\phi\sin\psi - \cos\phi\sin\theta\cos\psi \\ -\cos\theta\sin\psi & \sin\theta & \cos\theta\cos\psi \end{pmatrix}$$

Related topics

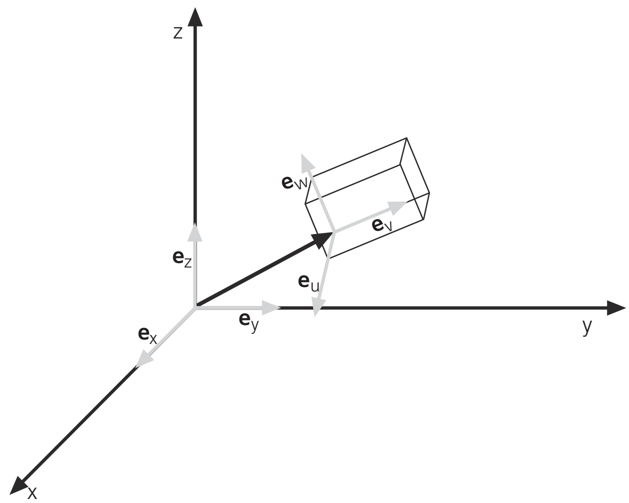
Basics

Coordinate System Used in MotionDesk.......... 13
Euler Angles.......... 15

3-D Objects in a 3-D Space

3-D objects

In contrast to a vertex, a 3-D object has an expansion, see the following illustration.



When defining a position for a 3-D object, you have to consider two different coordinate systems: the world and the local coordinate systems. Every 3-D object has its own local coordinate system.

In the illustration above, the world coordinate system is defined with the orthogonal unity vectors $\vec{e}_x$, $\vec{e}_y$, $\vec{e}_z$ and the local coordinate system is defined with the orthogonal unity vectors $\vec{e}_u$, $\vec{e}_v$, $\vec{e}_w$.

In general, the orientations of a local coordinate system and the world coordinate system are different.

Related topics

Basics

Working with the MotionDesk Blockset

Introduction

You can learn to use the MotionDesk Blockset to calculate and stream the motion data from the simulation to MotionDesk.

Where to go from here

Information in this section

Basics on Preparing the Simulation Model with the MotionDesk Blockset

Introduction

MotionDesk contains the **MotionDesk Blockset** to create Simulink models to calculate the motions of a simulation and send them to MotionDesk.

Signal values can also be displayed with MotionDesk instruments.

Using the MotionDesk Blockset

Introduction

You can use the **MotionDesk Blockset** to connect the simulation with MotionDesk for **MotionDesk simulations**. The blockset streams the calculated motion data to MotionDesk. The model can be built in MATLAB Simulink.

The **MotionDesk Blockset** supports all the simulation platforms listed in Blockset Supported Platforms on page 12.

You can learn to use the MotionDesk Blockset to adapt Simulink models to calculate and stream motion data in Adapting Simulink Models on page 24.

Related topics

Basics

Configuring Network Settings for Blockset Ethernet Connections

Introduction

To work with the **MotionDesk Blockset**, you must configure your network settings to communicate via Ethernet.

Where to go from here

Information in this section

Network Basics for the MotionDesk Blockset

Introduction

Provides basic network information to work with the MotionDesk Blockset, for example, TCP and UDP/IP protocols, IP address, private address space, UDP ports, and firewall settings.

TCP and UDP/IP protocols

The communication between PCs in a network is standardized by protocols. Like the widely known Transmission Control Protocol (TCP), the User Datagram Protocol (UDP) regulates the transport of data packages (datagrams) in a network in conjunction with the lower-level Internet Protocol (IP). In contrast to TCP, UDP does not guarantee that each data package reaches the receiver, but it is much faster and leads to shorter latency times than TCP. For online visualization, a lost datagram is no problem because the next one can be used.

The MotionDesk Blockset and MotionDesk uses UDP/IP because it provides the fast and efficient data transport which is necessary for real-time data exchange.

IP address

Every PC in a network – or more precisely, every network adapter or host – needs a unique address. UDP/IP uses 32-bit values for the host address, known as the IP address.

The IP address is typically written in 4 parts, separated by dots. Each part contains a number from 0 to 255 (one byte), and a typical network address is written as in the following example: 192.168.0.15.

The first one, two or three parts of the address identify the network (subnet address), the rest represents the host address itself. The parts used for the subnet address can be identified in the network mask. This mask resembles a host address. Each part of the subnet address is given the number 255, each part of the host address the number 0.

If the IP address of your host is 192.168.0.15 and the mask is 255.255.255.0, the subnet address is 192.168.0, and the host address is 15.

Private address space

In principle, you can use all possible numbers for IP addresses in a local network, as long as each host is given its own unique address.

However, this could cause problems if the local network is connected to the Internet sometime in the future.

For this reason it is advisable to use special addresses that are free for local networks and will not disturb any Internet connections. The following list shows the private address spaces:

- 10.0.0.0 to 10.255.255.255
- 172.16.0.0 to 172.16.255.255
- 192.168.0.0 to 192.168.255.255

Some numbers in the host part of the address are reserved for special purposes:

- If all host parts of the address are 0, this represents the entire network.
- If the host address is 255, you have the broadcast address, that is, all the hosts in the network receive the sent message. For example, 192.168.255.255 (network mask: 255.255.0.0) defines the broadcast address of subnet 192.168.
- Also reserved are the addresses 0.0.0.0 and 127.0.0.0 (network mask: 255.0.0.0). They define the so-called standard loop and loopback addresses.

UDP ports

The IP address specifies a specific host. Together with a port number, it defines a socket. This allows multiple processes to use UDP services on the same host for long-term conversations.

UDP provides port numbers from 1 to 65535. The numbers from 1025 to 65535 are not reserved and are free to use for your own network.

MotionDesk uses three UDP ports:

- Port 1 is used to send motion data from the simulation to the MotionDesk PCs
- Port 2 is used to send configuration requests from the MotionDesk PCs to the simulation, for example, to ask for the names of the 3-D objects.
- Port 3 is used to send configuration responses from the simulation to the MotionDesk PCs, for example, to send the names of the 3-D objects.

Firewall settings

To transfer data via Ethernet, the firewalls of the PCs must be configured to allow the communication for dSPACE software. As in a Simulink simulation data is transferred via Ethernet, the firewalls must allow the communication also for MATLAB.

Windows firewalls During the installation of dSPACE software, Windows firewalls are automatically configured to allow communication between the platform management instances for synchronization. You do not have to configure Windows firewalls manually. The first time you start a product involved in platform management synchronization, the firewall asks you to allow access. Confirm the product as trusted software.

Other firewalls If the host PC has a firewall different from the Windows firewall, configure that firewall manually to allow communication of dSPACE products and MATLAB via Ethernet.

How to Configure the Windows Network Settings for a Multi-PC Solution

Objective

If you have connected one or more additional MotionDesk PCs to the simulator, you first have to configure the Windows settings for the network adapters. Depending on your Windows operating system, the appropriate dialog can look slightly different. Refer to the documentation of your network adapter for details.

Method

To configure the Windows network settings for a multi-PC solution

1 In the Windows network settings dialog, configure the network adapter that is used for visualization in the MotionDesk PC, according to the following list:
   - Set the IP address for the network adapter.
     To build a local network, choose an address from the private address space (10.0.0.1 to 10.255.255.254, or 172.16.0.1 to 172.16.255.254, or 192.168.0.1 to 192.168.255.254).
   - Define the network mask (subnet mask), for example, **`255.155.255.0`**.

   You do not have to define values for Default Gateway, DNS, WINS address, or Routing to build up a local network.

Related topics

HowTos

How to Configure the Network Settings in MotionDesk

Objective

After configuring the network settings in Windows and the simulation model, you must also configure them in MotionDesk.

Preconditions

You configured the network settings in Windows and the simulation model.

Method

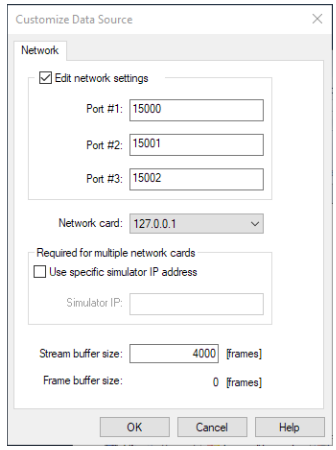
To configure the network settings in MotionDesk

1 In MotionDesk, select Network: UDP: Connection as the data source in the Motion Player.

2 In the Motion Player, click the Browse button.
   The Customize Data Source dialog opens.

3 Select Edit Network Settings and set the values according to the following list:
   - Set the network Port #1 to Port #3 to the same values as defined in the MD_Communication block.
   - In Network Card, select the IP address of the network adapter. If you have more than one network adapter in your MotionDesk PC, make sure you select the one used for your local visualization network.

The following illustration shows the dialog filled with typical values.



Result

The network settings in MotionDesk are configured and you can visualize the model running in a simulation.

Related topics

HowTos

Adapting Simulink Models

Introduction

You can create a Simulink model using the MotionDesk Blockset blocks to to calculate the motion data and stream to MotionDesk

Where to go from here

Information in this section

Information in other sections

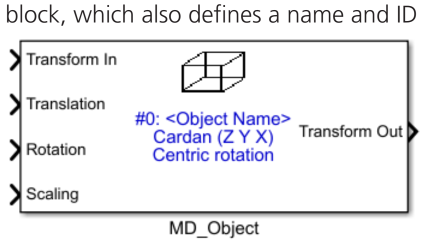
Calculating Motion Data for Movable Objects

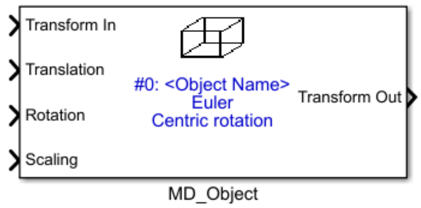
Introduction

To move the movable objects in MotionDesk, the simulation model must calculate the motion data.

MD_Object block

The motion data consists of the transformation matrices of all the movable objects in a model. A transformation matrix is calculated by an MD_Object block, which also defines a name and ID for the movable object.





The Transform In input defines an initial position for the movable object. To set the initial position to the origin of the coordinate system, you can use the MD_Inertial_Coordinate_System block. Starting from this position, the object is moved by the Translation and Rotation vector to the new position, which is specified by the Transform Out port. With this method, it is possible to define kinematic chains. Refer to Building Kinematic Chains on page 26.

The MD_Object block allows you to specify the sequence of translation and rotation, and the formats of the rotation angles. For more information on this block, refer to MD_Object on page 60.

Tip

You can also use the MD_Object block if you want to calculate the transformation matrix without creating a movable object.

Related topics

References

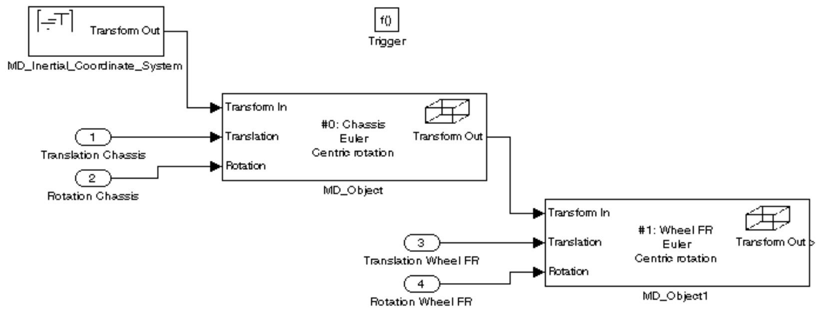
Building Kinematic Chains

Introduction

You can create kinematic chains using the MotionDesk Blockset blocks. A kinematic chain defines dependencies between the movements of movable objects. For example, if a car moves in a virtual world, the chassis is moved relative to the world, but the wheels are moved relative to the chassis.

Example

The following illustration shows an example of a kinematic chain. For the sake of clarity, it contains only the chassis and one wheel.



The translation and rotation of the chassis are relative to the world. Therefore, the input transformation matrix of the chassis is given by the MD_Inertial_Coordinate_System block. As the wheel is connected to the chassis, the wheel's translation and rotation are relative to the chassis. Therefore, the input transformation matrix of the wheel is given by the output transformation matrix of the chassis.

Related topics

References

MD_Inertial_Coordinate_System... 66
MD_Object... 60

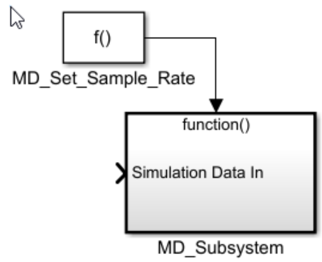
How to Define a Sampling Rate for the Animation

Setting the sample time

Generally, the step size of the simulation is too fast to display every step in MotionDesk. In some cases it is unnecessary to calculate the motion data with the same step size as the simulation. However, if a low latency time is required, for example, for a man-in-the-loop application, a short sample time is strongly recommended. The sample time depends on the use scenario, for example, in a man-in-the-loop scenario a sample time of 10 ms, which results in 100 frames per second, is sufficient. The step size of the simulation, however, must still be about 1 ms. In an automated maneuver a sample time of 20 ms may be sufficient.

MotionDesk subsystem

The sample time of the animation can be controlled by integrating the MotionDesk Blockset blocks in a subsystem and triggering it accordingly. All MotionDesk Blockset blocks must be placed in the same subsystem.



Method

To define a sampling rate for the animation

1 Place the 3-D_Subsystem template (MD_Set_Sample_Rate block and MD_Subsystem) in the model.

2 Double-click the MD_Set_Sample_Rate block and set the Sample time according to your needs.

3 Double-click the MD_Subsystem to open it.

4 In the MD_Subsystem, place the MotionDesk Blockset blocks that process the motion data and connect them. For information on how to do this, see Calculating Motion Data for Movable Objects on page 25.

5 In the MD_Subsystem, place the communication block for the data transfer to MotionDesk. For information on how to do this, see How to Provide MotionDesk with Motion Data on page 28.

6 Close the MD_Subsystem.

7 Connect the signals of the simulation model to the MD_Subsystem.

Result

The motion data is calculated and transferred in the sampling rate specified by the Sample time parameter of the MD_Set_Sample_Rate block.

Related topics

References

How to Provide MotionDesk with Motion Data

Objective

One MotionDesk block must be inserted in the real-time model to control the data transfer from the simulation to MotionDesk.

Basics

One block is needed to transmit motion data to MotionDesk. The MD_Communication block collects all motion data from the MD_Object blocks of the model and generates data packages which are transmitted to MotionDesk.

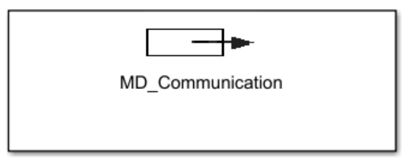
Multi-PC solution or Simulink simulation

To set up communication via Ethernet in a multi-PC solution or a Simulink simulation, you must specify the network parameters. For basic information on the network parameters, refer to Setting Up a Multi-PC Solution on page 34 or Configuring Network Settings to Visualize a Simulink Simulation on page 43.

Method

To transmit motion data

1 Place the MD_Communication block in the MD_Subsystem of your real-time model.

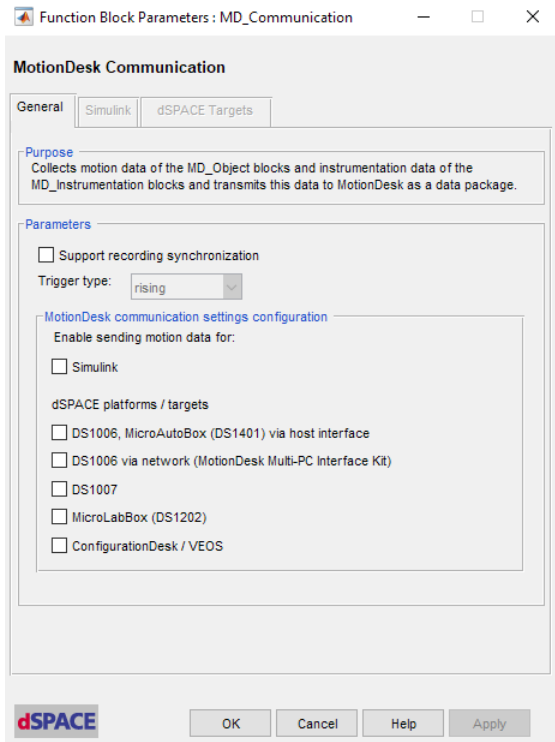


MD_Communication

2 Open the block dialog.

3 On the General page, specify the parameters and select the platform. For details on the dialog settings, refer to General Page (MD_Communication) on page 54.

Depending on the selected platform, the **Simulink** page and/or **dSPACE Targets** page is enabled.



4 Specify the parameters that are specific for the platform:

- If Simulink is selected, open the **Simulink** page and specify the parameters. For details of the parameters, refer to Simulink Page (MD_Communication) on page 56.
- If a dSPACE platform or target is selected, open the **dSPACE Targets** page and specify the parameters. For details of the parameters, refer to dSPACE Targets Page (MD_Communication) on page 57.

Result

The motion data is transferred from the simulation application to the MotionDesk PC during run time.

Related topics

References

How to Prepare the Real-Time Model for Using Instruments or State Objects

Objective

To use instruments to display values of simulation variables or state objects in MotionDesk, the simulation model must be extended.

Basics of instruments

For basic information on using instruments in MotionDesk, refer to Basics on Using Instruments in the Scene (MotionDesk Scene Animation 📖).

Basics of state objects

State objects are 3-D objects that have subobjects which can change their look according to a value of a simulation variable. For details of state objects and the possible values of the simulation variable, refer to Basics of Using State Objects in the Scene (MotionDesk Scene Animation 📖).

Preparing the real-time model

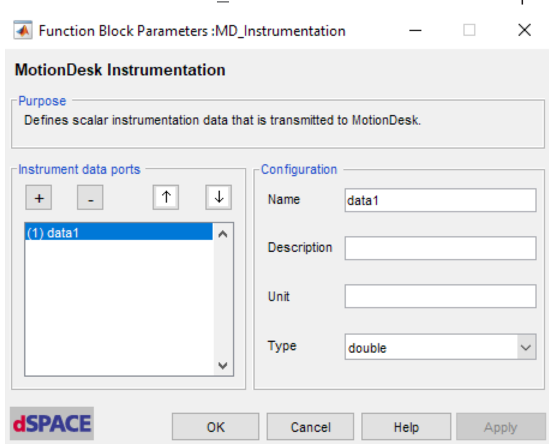
The instrument data and the states of the state objects are transferred from the real-time model to MotionDesk by the MD_Instrumentation block. The block is contained in the MotionDesk Blockset. To send data for instruments, you have to place the MD_Instrumentation block in your real-time model and connect the simulation variables to it.

Method

To prepare the real-time model for using instruments or state objects

1 Place the MD_Instrumentation block in the MD_Subsystem of your real-time model.

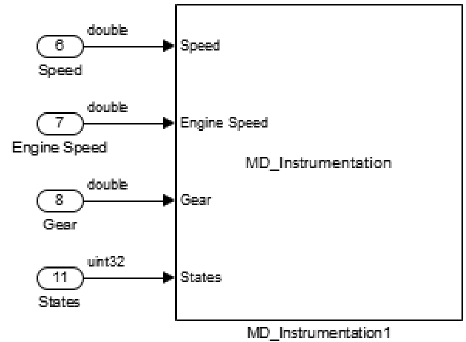
2 Double-click the MD_Instrumentation block to open the block dialog.



3 In the block dialog, specify the number of data ports (variables to be sent) and the properties of the variables.

4 Click OK to close the block dialog.

5 Connect the simulation variables to the inports of the MD_Instrumentation block.



6 Save your real-time model.

Result

The real-time model is prepared for transferring the values of variables for the instruments. Now you can build the simulation application and download it to the simulation platform as usual to start the simulation.

Related topics

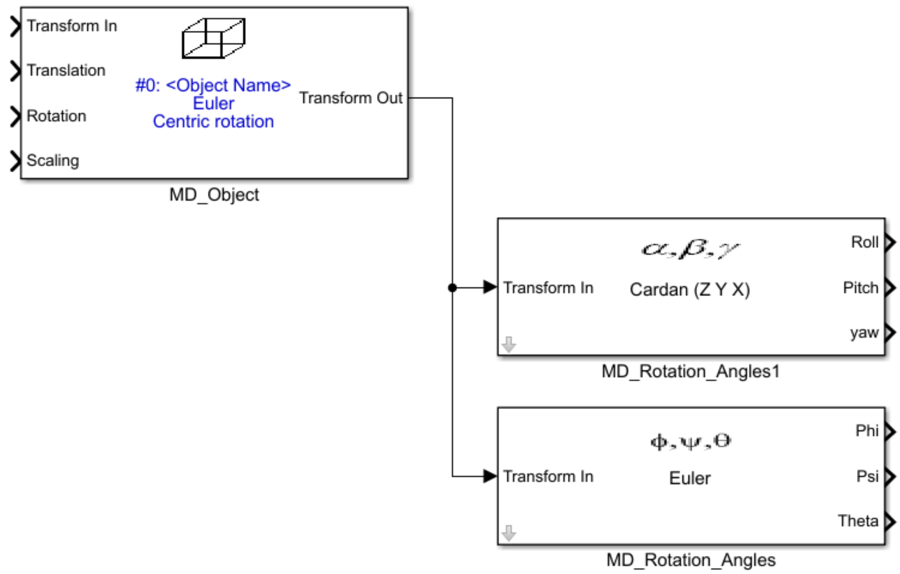
References

MD_Instrumentation..70

How to Observe the Rotation Angles

Objective

A transformation matrix is not suitable for interpreting a rotation. A rotation of a 3-D object can best be observed when it is displayed in 3 separate rotation angles. These angles can be calculated by a special block, see the following illustration.



Angle interpretation system

The MD_Rotation_Angles block can output the angles in two different formats: the Euler and cardan (roll-pitch-yaw) angle interpretation system. For details of the angle systems, refer to Euler Angles on page 15 or Cardan Roll, Pitch, and Yaw Angles on page 16.

Method

To observe the rotation angles

1 Place an MD_Rotation_Angles block in the MD_Subsystem of your real-time model.
2 Open its block dialog and select the angle interpretation system: Euler or cardan (roll-pitch-yaw).
3 Connect the input port to a Transform Out port of an MD_Object block.
4 Connect the output ports of the MD_Rotation_Angles block to an appropriate Simulink block, for example, Simulink's Constant blocks.

Result

You can observe the rotation angles with your experiment software.

Related topics

Basics

References

Setting Up a Multi-PC Solution

Introduction

If you want to visualize multiple views of a scene on separate displays, you have to share the simulation and visualization tasks. One PC controls and monitors the simulation, and other PCs perform the visualization tasks.

Where to go from here

Information in this section

Basics of Setting Up a Multi-PC Solution Using the MotionDesk Blockset

Introduction

In a multi-PC solution, one platform provides the motion data for several MotionDesk PCs. This feature allows to view the same scene from different viewpoints.

Required dSPACE hardware

A multi-PC solution is possible with the following RTI and ConfigurationDesk / VEOS supported platforms:

RTI Platforms

- DS1006
- DS1007
- MicroLabBox

ConfigurationDesk platforms / VEOS

- SCALEXIO
- DS6001
- MicroAutoBox III
- VEOS

For a DS1006 platform you have to install the MotionDesk Multi-PC Interface Kit on the dSPACE Simulator.

The Multi-PC Interface Kit, the SCALEXIO system, DS1007, MicroLabBox, and VEOS can be connected with one or more additional PCs via Ethernet.

Configuring the network settings

After connecting the hardware, you have to configure it. This means that you have to specify the port numbers and IP addresses in the simulation model for DS1006 platforms (MotionDesk Blockset) and on the MotionDesk PCs.

Connecting Hardware for a Multi-PC Solution

Introduction

A Multi-PC Interface Kit is required for a multi-PC solution with DS1006-based real-time systems. Additional hardware is not required for a multi-PC solution in combination with MicroLabBox and DS1007 or with the ConfigurationDesk supported platforms, SCALEXIO, DS6001, MicroAutoBox III, and VEOS.

In a multi-PC solution, motion data is sent from the simulator to the MotionDesk PCs via an Ethernet network.

⚠ WARNING

It is strongly recommended to use a dedicated local network for your broadcast-based visualization tasks. Do not use the company LAN. The broadcast mode for sending messages produces an enormous amount of network traffic, which may seriously impede network operation. If you need to have a connection to your company LAN on the MotionDesk PCs, use two separate network adapters on them, one for the company LAN and one for your local visualization network.

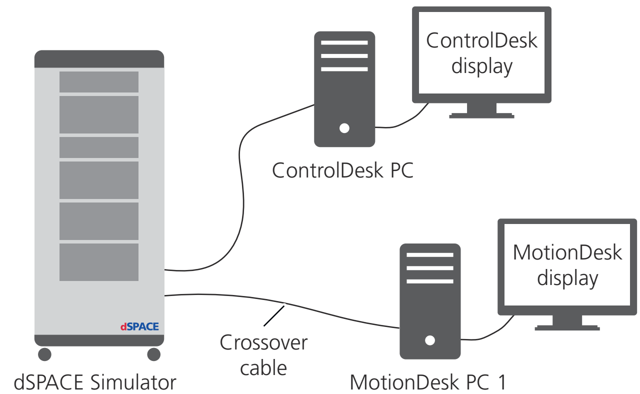
DS1006-based systems

If you want to set up a multi-PC solution, you must install the MotionDesk Multi-PC Interface Kit on the simulator based on a DS1006 Processor Board. The interface board needs a free 8-bit ISA slot and is connected to the DS1006 Processor Board via a PHS bus cable. You can connect one or more PCs to the interface board via Ethernet.

To build a small local Ethernet network, you have to install Ethernet network adapters on the additional PCs (called MotionDesk PCs below) and connect them and the interface board to a hub. The MotionDesk Multi-PC Interface Kit has a bandwidth of 10 Mbit/s. If you use a 100 Mbit/s hub, it must be compatible with 10 Mbit/s interfaces. However, it should not cause problems to use 100 Mbit/s network adapters for the MotionDesk PCs.

One additional PC

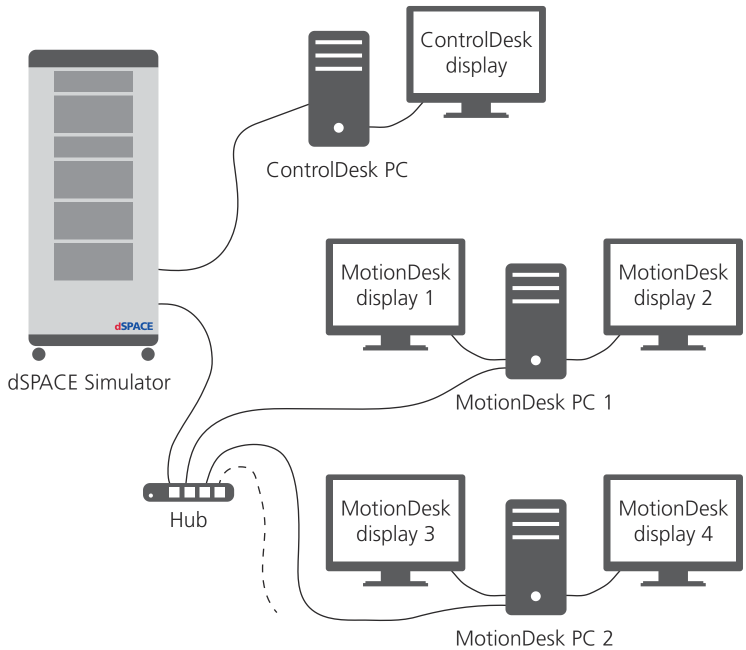
If you want to connect only one additional PC to the simulator, you do not need a hub. You can setup a direct connection with a crossover cable, as shown in the illustration below.



This solution allows you to use separate displays for ControlDesk and MotionDesk.

More additional PCs

To connect more than one PC to the simulator, you need a hub. The illustration below shows 2 MotionDesk PCs connected to a hub. Depending on the size of the hub, you can connect more PCs to the simulator. You can extend the number of hubs by cascading them, that is, connecting one hub to another.



This solution enables you not to have only separate displays for ControlDesk and MotionDesk, but also separate displays for different views of a MotionDesk scene with very high performance.

For example, you can set up different driver views and place them on separate displays. This allows you to use MotionDesk in professional driving simulators, where you need front, side and rear views on separate displays.

Related topics

Basics

Multi-PC Solution (MotionDesk Basics 📖)

Workflow for Configuring a Multi-PC Solution

Introduction

After installing the network hardware, you have to configure it. To set up a small local network is very easy. You can do this without knowing much about networking.

Basics of network settings

If you are not familiar with network settings, it is advisable to read the basic information on network protocols, addresses, and ports. Refer to Network Basics for the MotionDesk Blockset on page 21.

Basics

If you have connected one or more additional MotionDesk PCs to the simulator, you have to configure the network settings in the Windows network settings dialog, in your simulation model, and in the data source settings in MotionDesk.

> **Note**
>
> To change network settings like addresses or ports, you need to be logged on as an administrator.
> It is assumed that you set up a small dedicated network for this visualization task and do not use the company LAN. However, if you want to use the company LAN, ask the system administrator for the network settings.

The following workflow describes the configuration of a multi-PC solution. If one PC is used for visualization, the broadcast mode is not used. If you need a higher data rate than 50 data packages per second or want to use several MotionDesk PCs, you must use the broadcast mode in a dedicated network.

> **Note**
>
> Broadcast mode is not used by the installer as only TCP connections are established.

Workflow for configuring a multi-PC solution

To configure a multi-PC solution, you must perform the following steps:

1. First, you have to configure the Windows settings for the network adapters. Refer to How to Configure the Windows Network Settings for a Multi-PC Solution on page 22.
2. The Windows settings for the network adapters determine the address parameters in the communication block of your simulation model. Refer to How to Configure the Network Settings in the Simulation Model on page 39.
3. Finally, you have to tell MotionDesk that you will use a network data source and what settings it has. Refer to How to Configure the Network Settings in MotionDesk on page 23.

Related topics

Basics

Network Basics for the MotionDesk Blockset..21

How to Configure the Network Settings in the Simulation Model

Objective

After configuring the Windows network settings, you must configure the network settings in the simulation model as well.

Precondition

You configured the network settings in Windows.

Possible methods

The configuration of the network settings in the simulation model depends on the platform type and the number of additional PCs that are connected to the platform:

- DS1006-based systems and *one* additional PC. Refer to Method 1 on page 39.
- DS1006-based systems and *more than one* additional PC. Refer to Method 2 on page 40.
- DS1007-based systems, MicroLabBox or ConfigurationDesk supported platforms: SCALEXIO, DS6001, MicroAutoBox III, and VEOS. Refer to Method 3 on page 41.

Method 1

To configure the network settings in the simulation model of a DS1006-based system and one additional PC

1. In the simulation model, double-click the MD_Communication block.
2. On the dSPACE Targets page, set the parameters according to the following list:
   - In dSPACE platforms / targets, select DS1006 via network (MotionDesk Multi-PC Interface Kit).
   - If you have installed more than one MotionDesk Multi-PC Interface Kit, select the appropriate Board number, otherwise select 1.
   - Set the Simulator IP address according to your local visualization network.

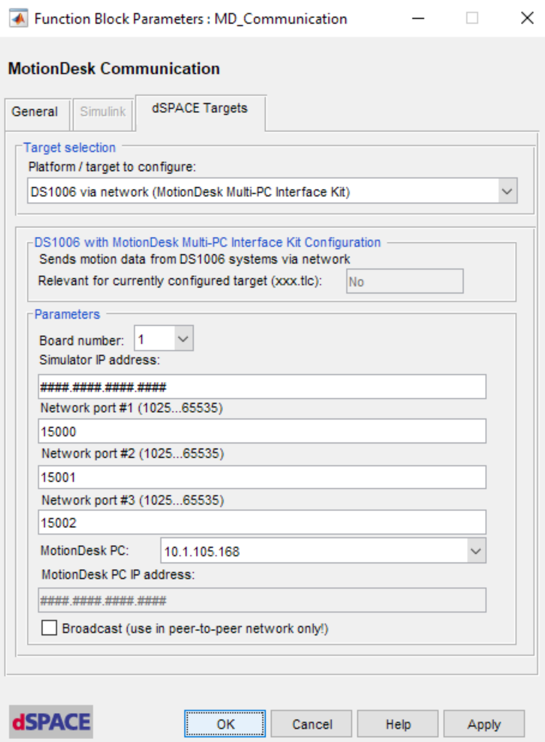
     Make sure that the simulator belongs to the same network as the MotionDesk PC. The subnet part of the address must be the same as in the MotionDesk PC IP address, the host part must be unique.
   - Set the Network port #1 to #3 to different values between 1025 and 65535, or use the default settings.
   - From the MotionDesk PC list, select Remote MotionDesk PC.

     The MotionDesk PC IP address edit field is enabled.
   - Set the MotionDesk PC IP address according to the settings of the network adapter in the MotionDesk PC.

If you have more than one network adapter in your MotionDesk PC, make sure you select the one used for your local visualization network.

- Do not select the Broadcast checkbox.

The following illustration shows the dialog filled with typical values.



Method 2

To configure the network settings in the simulation model of a DS1006-based system and more than one additional PC

1 In the simulation model, double-click the MD_Communication block.

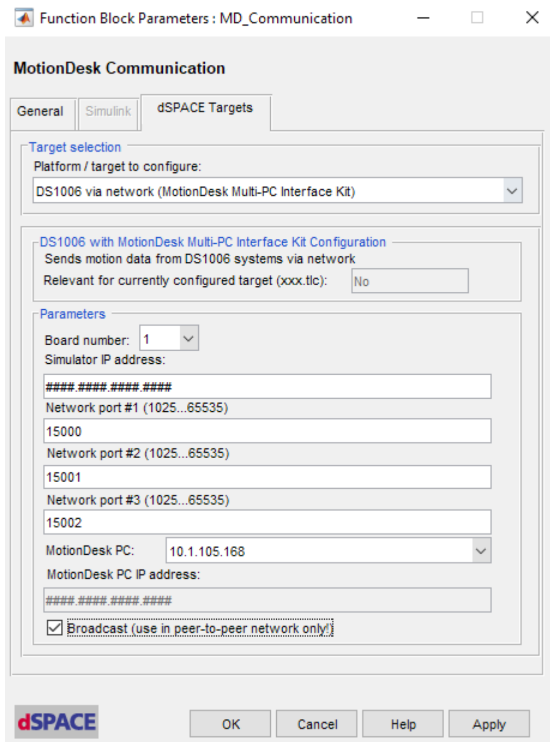
2 On the dSPACE Targets page, set the parameters according to the following list:

- In dSPACE platforms / targets, select DS1006 via network (MotionDesk Multi-PC Interface Kit).
- If you have installed more than one MotionDesk Multi-PC Interface Kit, select the appropriate Board number, otherwise select 1.
- Set the Simulator IP address according to your local visualization network.

  Make sure that the simulator belongs to the same network as the MotionDesk PCs. The subnet part of the address must be the same as in the MotionDesk PC IP addresses, the host part must be unique.
- Set the Network port #1 to #3 to different values between 1025 and 65535, or use the default settings.
- From the MotionDesk PC list, select the MotionDesk PC IP address according to the settings of the network adapter in one of the MotionDesk PCs, or take a dummy address for it.

Do not leave this edit field empty. Although MotionDesk ignores the value, the dialog cannot be closed without an entry.

- Select the Broadcast checkbox.



Method 3

To configure the network settings in the simulation model of a DS1007, MicroLabBox platform or ConfigurationDesk supported platforms: SCALEXIO, DS6001, MicroAutoBox III, and VEOS.

1. In the simulation model, double-click the MD_Communication block.
2. On the dSPACE Targets page, set the parameters according to the following list:
   - In dSPACE platforms / targets, select DS1007, MicroLabBox (DS1202), or ConfigurationDesk / VEOS.
   - To send motion data to one PC, clear the Broadcast checkbox.
   - To send motion data to several PCs, select the Broadcast checkbox.

Result

The network settings in the simulation model are configured according to the multi-PC solution architecture.

Next step

How to Configure the Network Settings in MotionDesk on page 23.

Related topics

HowTos

Setting Up Visualization for a Simulink Simulation

Introduction

You can visualize a PC-based Simulink simulation on 1 or 2 PCs. You use the same network services as in a multi-PC solution in both cases. Therefore the PC must have a network adapter, even if it is not connected to a network.

Where to go from here

Information in this section

Information in other sections

System Overview (MotionDesk Basics 📖)
Provides an overview of the different visualization systems.

Connecting Hardware to Visualize a Simulink Simulation

Introduction

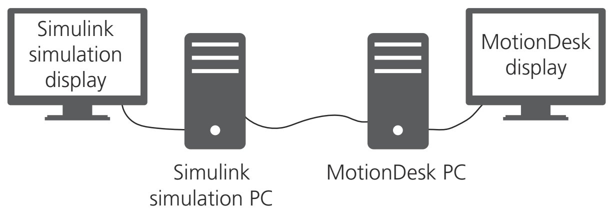
As the Simulink simulation runs without real-time hardware, you do not have to connect a simulator or a MotionDesk Multi-PC Interface Kit. It is sufficient to install a network adapter. If you use 2 PCs, you have to connect them via Ethernet.

1 PC

Even if you want to use only 1 PC, a network adapter must be installed because network services are used to transmit motion data. The PC must not be connected to a network.

2 PCs

If you use 2 PCs, you have separate displays for Simulink and MotionDesk. Additionally, you increase the performance for large simulation models. The 2 PCs are connected directly via a crossover cable (peer-to-peer connection) or they are connected as hosts in a LAN. The illustration below shows a peer-to-peer connection.



Related topics

Basics

Simulink Simulation (MotionDesk Basics 📖)

Configuring Network Settings to Visualize a Simulink Simulation

Introduction

After connecting the hardware, you have to configure it. This means that you have to specify port numbers and IP addresses.

Workflow for configuring the network settings

The following steps describe the case of building a peer-to-peer network between 2 PCs. One PC is used for Simulink simulation, the other for MotionDesk.

If you use only 1 PC, use its IP address whenever you have to select an IP address. The network adapter will be used as sender and receiver simultaneously.

Note

To change network settings like addresses or ports, you need to be logged on as an administrator.
It is assumed that you set up a small dedicated network for this visualization task and do not use the company LAN. However, if you want to use the company LAN, ask the system administrator for the network settings.

In the Simulink simulation, the maximal frame rate is limited to 50 frames per second, to avoid disturbance in company LANs.

You have to configure the network settings in the Windows network settings dialog, in your simulation model and in the data source settings in MotionDesk. Perform the following steps:

1. First you have to configure the Windows settings for the network adapters. Refer to How to Configure the Windows Network Settings for a Multi-PC Solution on page 22.
2. The Windows settings for the network adapters determine the address parameters in the communication block of your simulation model. Refer to How to Configure the Network Settings in the Simulation Model for a Simulink Simulation on page 44.
3. Finally, you have to tell MotionDesk that you will use a network data source and what settings it has. Refer to How to Configure the Network Settings in MotionDesk on page 23.

Related topics

Basics

How to Configure the Network Settings in the Simulation Model for a Simulink Simulation

Objective

After configuring the Windows network settings, you must configure the network settings in the simulation model as well.

Precondition

You configured the network settings in Windows. Refer to How to Configure the Windows Network Settings for a Multi-PC Solution on page 22.

Method

To configure the network settings for a visualization based on a Simulink simulation

1 In the simulation model, double-click the MD_Communication block.

2 On the Simulink page, set the parameters as follows:

- Select the Local PC IP address of the Simulink PC.

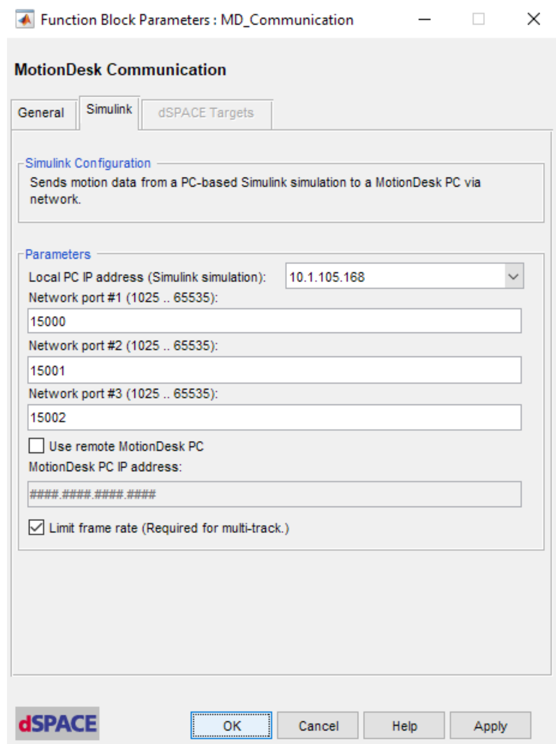
  If you use only one PC, the MotionDesk PC IP address and the Local PC IP address have to be equal.
- Set the Network port #1 to #3 to different values between 1025 and 65535, or use the default settings.

- Select Use remote MotionDesk PC if you work with more than one PC.

  The MotionDesk PC IP address edit field is enabled.
- Set the MotionDesk PC IP address according to the settings of the network adapter in the MotionDesk PC.

  If you have more than one network adapter in your MotionDesk PC, make sure you select the one used for your local visualization network.

  If you use only one PC, the MotionDesk PC IP address and the Local PC IP address are the same.
- Select Limit frame rate to adjust the frame rate for using the multi-track mode.



Result

MotionDesk can receive the motion data from a Simulink simulation.

Related topics

References

MD_Communication.. 52

Working with the MotionDesk Service Library

Introduction

You can learn to use the MotionDesk Service Library to calculate and stream the motion data from the simulation to MotionDesk. You must adapt handcoded models in C programming language with MotionDesk specific code in the real-time model.

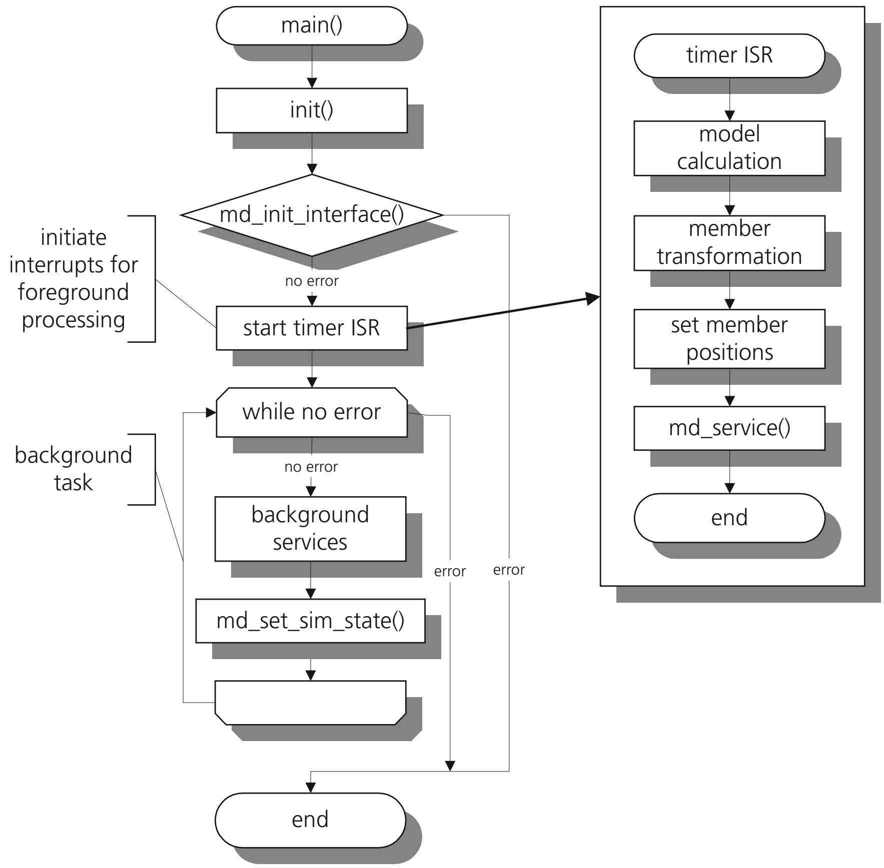
Adapting Handcoded Models

Introduction

You can learn to use the MotionDesk Service Library to calculate and stream the motion data from the simulation to MotionDesk. You must adapt handcoded models in C programming language with MotionDesk specific code in the real-time model.

MotionDesk Service Library

MotionDesk service library C functions must be integrated in the handcoded model. Model computation, member transformation and host synchronization have to be performed subsequently in the model foreground. The following illustration shows the flow chart for the MotionDesk service mechanism:



For a description of the MotionDesk service functions, refer to MotionDesk Service Library on page 75.

Reference Information

Where to go from here

Information in this section

MotionDesk Blockset

Introduction

Provides reference information on the blocks of the MotionDesk Blockset.

Where to go from here

Information in this section

General Information on the MotionDesk Blockset

Introduction

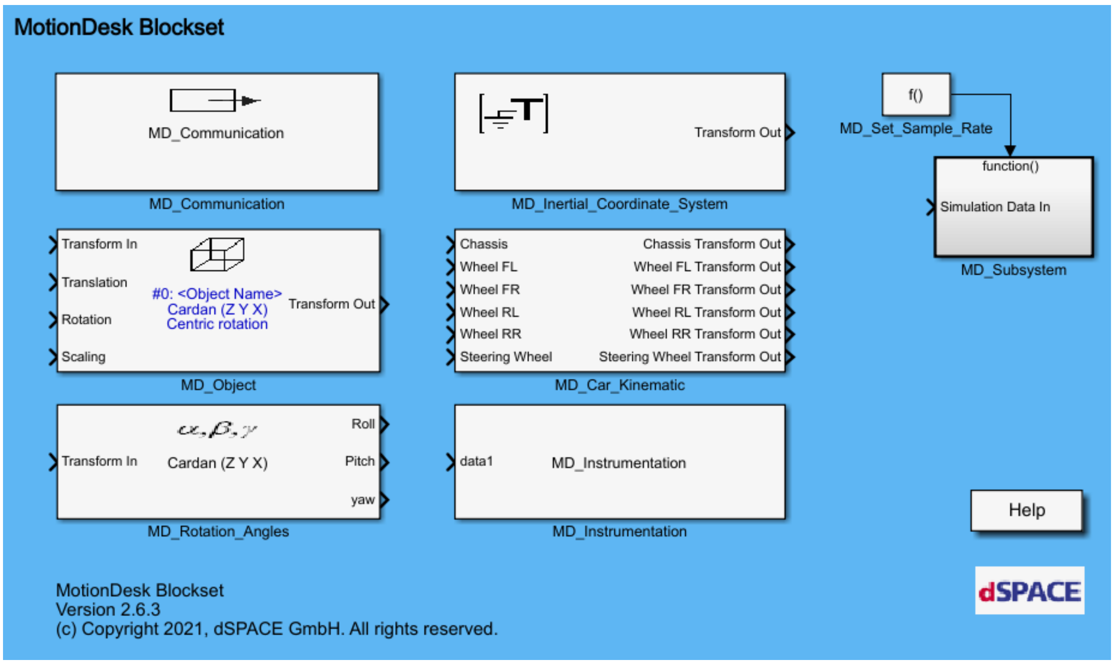
Provides an overview of the MotionDesk Blockset.

Overview of the MotionDesk Blockset

Library access

To open the MotionDesk Blockset, type **mdbs** in the MATLAB® command window.

You can also open the blockset via the DS1006, DS1007, MicroLabBox, or MicroAutoBox II blockset by clicking: Blocksets – MotionDesk Blockset.



Library components

The following blocks are available in the MotionDesk Blockset:

- MD_Communication
- MD_Object
- MD_Rotation_Angles
- MD_Inertial_Coordinate_System
- MD_Car_Kinematic
- MD_Instrumentation
- MD_Set_Sample_Rate and MD_Subsystem

Rapid Accelerator mode

The blocks of the MotionDesk Blockset can be used in a Simulink simulation in the Rapid Accelerator mode. The following limitations apply:

- You need a C/C++ compiler to build MEX files.

  For more information on the supported software and versions, refer to C/C++ compiler for building MEX files (nonTargetLink) in Required C and C++ Compilers (Installing dSPACE Software).

- Only one simulation must be executed at the same time, regardless of which simulation mode is used.

MotionDesk Blockset supported platforms

MotionDesk and the MotionDesk Blockset support the following platforms:

- Simulink
- SCALEXIO Systems
- MicroAutoBox III
- VEOS and VEOS on Linux

RTI platforms

- DS1006
- DS1007
- MicroLabBox (DS1202)
- MicroAutoBox II (DS1401)

Note

To work with Sensor Simulation, you must use the Model and Sensor Interface Blockset. For more information, refer to Basics of the Model and Sensor Interface Blockset (Model and Sensor Interface Blockset Manual 🕮).

Related topics

References

MD_Communication

Purpose

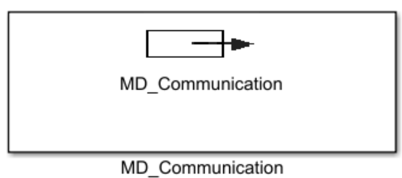
To automatically collect all motion data from different MD_Object blocks and instrumentation data from a MD_Instrumentation block and transmit them to MotionDesk.

Where to go from here

Information in this section

Block Description (MD_Communication)

Illustration



Purpose

To automatically collect all motion data from different MD_Object blocks and instrumentation data from a MD_Instrumentation block and transmit them to MotionDesk.

Description

The MD_Communication block must be used to configure the data transfer of motion data from the model to MotionDesk. It collects motion data calculated by all the MD_Object blocks and instrumentation data from the MD_Instrumentation block and transfers it to MotionDesk. Only one MD_Communication block must be used in the model.

The block handles the communication of the motion and instrumentation data:

- Data calculated in a Simulink simulation (Normal mode, Accelerator mode, or Rapid Accelerator mode) and transferred via network to MotionDesk (on the same or another PC).
- Data calculated in a real-time simulation and transferred to MotionDesk via host interface (possible for DS1006 and MicroAutoBox).

- Data calculated in a real-time simulation and transferred to one ore more MotionDesk PCs via Ethernet (possible for DS1006 with the MotionDesk Multi-PC Interface Kit, DS1007, and ConfigurationDesk supported platforms: SCALEXIO, DS6001, MicroAutoBox III and VEOS). For more information, refer to Setting Up a Multi-PC Solution on page 34.

Note that the usage of the MD_Communication block, MD_Instrumentation block and MD_Object blocks in the model is limited:
- All used blocks must have the same sample time.
- All used blocks must be in the same atomic subsystem.

I/O characteristics

The table shows the block inport:

Simulink Input	Simulink Data Type	Meaning
Synchronization	Double	The inport is used to synchronize motion data in the multi-track mode. It must be enabled on the General page[1].

[1] For details, refer to General Page (MD_Communication).

Dialog pages

The dialog settings can be specified on the following pages:
- General Page (MD_Communication) on page 54
- Simulink Page (MD_Communication) on page 56
- dSPACE Targets Page (MD_Communication) on page 57

Related topics

References

MD_Object.. 60

General Page (MD_Communication)

Purpose

To specify basic settings for the communication (data transfer of the motion data to MotionDesk).

Description

In the dialog you specify the parameters for the support of synchronized recording and selects how motion data is transferred to MotionDesk. The block supports different communication methods. The communication via host interface and MotionDesk Multi-PC Interface Kit is not possible at the same time.

Dialog settings

Support recording synchronization Indicates whether recording synchronization is enabled. If checked, the Synchronization inport is created and the trigger type must be specified. An active signal at the inport resets the time stamp of motion data. This is necessary if you want to compare several recorded tracks. Refer to Comparing a Simulation With Recorded Tracks (MotionDesk Scene Animation 📖).

Trigger type Lets you specify the trigger type for the synchronization. The recording can be triggered by a rising or falling edge or either of them.

MotionDesk communication settings configuration Lets you select the platform or target that calculates the motion data and the connection type.

dSPACE targets	Description
Simulink	The motion data is calculated in a Simulink simulation. A network adapter is used to transfer the motion data. Refer to Simulink Simulation (MotionDesk Basics 📖).
DS1006, MicroAutoBox (DS1401) via host interface	The motion data is calculated on a DS1006 or DS1401 platform and transferred via the host interface. The platform is connected to the MotionDesk PC via a bus or network connection. Note that a network connection allows only a limited frame rate.
DS1006 via network (MotionDesk Multi-PC Interface Kit)	The motion data is calculated on a DS1006 platform. The motion data is transferred by the MotionDesk Multi-PC Interface Kit. This kit contains an I/O board that is installed in the modular system and transfers the motion data via a network connection.
DS1007	The motion data is calculated on a DS1007 and transferred via Ethernet.
MicroLabBox (DS1202)	The motion data is calculated on MicroLabBox and transferred via Ethernet.
ConfigurationDesk / VEOS	The motion data is calculated on the selected hardware platform or VEOS and transferred via Ethernet. Hardware platforms that are supported by ConfigurationDesk include: SCALEXIO, DS6001 and MicroAutoBox III.

If Simulink is selected in the General tab, the Simulink tab is enabled to specify the communication parameters.

If a value is selected in the dSPACE platforms / targets checkbox group, the dSPACE Targets tab is enabled to specify the communication parameters.

Related topics

Basics

Comparing a Simulation With Recorded Tracks (MotionDesk Scene Animation 🕮)
Simulink Simulation (MotionDesk Basics 🕮)

References

Simulink Page (MD_Communication)

Purpose

To specify the settings for a communication of a Simulink simulation to MotionDesk. The settings are valid for Normal mode, Accelerator mode, and Rapid Accelerator mode.

Dialog settings

Local PC IP address (Simulink simulation) Lets you choose the IP address of the PC which runs the Simulink simulation. If you use only one PC, the IP address of the MotionDesk PC and the local PC are the same.

Network port #1 (1025 .. 65535) Lets you specify the number of the UDP port that is used to send motion data from the simulator to the MotionDesk PCs. Each port must have a unique number between 1025 and 65535.

Network port #2 (1025 .. 65535) Lets you specify the number of the UDP port that is used to send configuration requests from the MotionDesk PCs to the simulator, for example, to ask for names of the 3-D objects. Each port must have a unique number between 1025 and 65535.

Network port #3 (1025 .. 65535) Lets you specify the number of the UDP port that is used to send configuration responses from the simulator to the MotionDesk PCs, for example, to send the names of the 3-D objects. Each port must have a unique number between 1025 and 65535.

Use remote MotionDesk PC Lets you choose which PC is used as the MotionDesk PC. If the checkbox is selected, the motion data is sent to the PC specified in the **MotionDesk PC IP address** edit field. Otherwise the local PC is used as the MotionDesk PC.

MotionDesk PC IP address Lets you specify the IP address of the MotionDesk PC.

Limit frame rate (Required for multi-track) Indicates whether the transmission rate of frames is adjusted. Limiting the frame rate is mandatory for the multi-track mode. Refer to Comparing a Simulation With Recorded Tracks (MotionDesk Scene Animation 🕮).

It slows down a Simulink simulation if MD_Communication is called more than 50 times per second. This ensures that MD_Communication transmits all data to MotionDesk.

Always select Limit frame rate when working with the multi-track mode in MotionDesk. Simulink then sends a signal to MotionDesk that the data is acquired with lossless data acquisition and therefore is track safe.

Related topics

Basics

Configuring Network Settings to Visualize a Simulink Simulation 43
Setting Up Visualization for a Simulink Simulation 42

dSPACE Targets Page (MD_Communication)

Purpose

To specify the settings for the communication of a real-time platform, VEOS or ConfigurationDesk.

Target selection

Lets you select the platform or target that will be used. When a target is selected, the block dialog shows the parameters that are relevant for the communication to the selected platform or target.

- DS1006 or MicroAutoBox via host interface
- DS1006 via network (MotionDesk Multi-PC Interface Kit)
- DS1007
- MicroLabBox
- ConfigurationDesk / VEOS

> **Note**
>
> You can select this if you use a hardware platforms that is supported by ConfigurationDesk or for VEOS.
> The hardware platforms that are supported by ConfigurationDesk are: SCALEXIO, DS6001 and MicroAutoBox III.

DS1006 or MicroAutoBox Configuration

Relevant for currently configured target (xxx.tlc) (read-only) Indicates whether sending motion data from the simulation platform to the MotionDesk PC is relevant for the currently configured target (xxx.tlc).

DS1006 with MotionDesk Multi-PC Interface Kit Configuration

Relevant for currently configured target (xxx.tlc) (read-only) Indicates whether sending motion data from the simulation platform to the MotionDesk PC is relevant for the currently configured target (xxx.tlc).

Board Number Lets you select the board number of the MotionDesk Multi-PC Interface Kit.

Simulator IP address Lets you specify the IP address of the simulator for a Multi-PC solution.

Network port #1 (1025 .. 65535) Lets you specify the number of the UDP port that is used to send motion data from the simulator to the MotionDesk PCs. Each port must have a unique number between 1025 and 65535.

Network port #2 (1025 .. 65535) Lets you specify the number of the UDP port that is used to send configuration requests from the MotionDesk PCs to the simulator, for example, to ask for names of the 3-D objects. Each port must have a unique number between 1025 and 65535.

Network port #3 (1025 .. 65535) Lets you specify the number of the UDP port that is used to send configuration responses from the simulator to the MotionDesk PCs, for example, to send the names of the 3-D objects. Each port must have a unique number between 1025 and 65535.

MotionDesk PC Lets you select the MotionDesk PC. The drop-down list displays all local IP addresses and the Remote MotionDesk PC option. Selecting Remote MotionDesk PC enables the MotionDesk PC IP address edit field.

MotionDesk PC IP address Lets you specify the IP address of the MotionDesk PC. This edit field is required if there is only one additional MotionDesk PC in the Multi-PC solution.

Broadcast (use in peer-to-peer network only!) Indicates whether the broadcast mode is used to send the network data packages. If the checkbox is selected, the data packages are sent to all hosts that are connected to the network.

Note

It is highly recommended to use a dedicated local network for your broadcast-based visualization tasks. Do not use the company LAN. The broadcast mode for sending messages produces an enormous amount of network traffic which may seriously impede network operation. If you need a connection to your company LAN on the MotionDesk PCs, use two separated network adapters on them, one for the company LAN and one for your local visualization network.

DS1007 Configuration

Relevant for currently configured target (xxx.tlc) (read-only) Indicates whether sending motion data from the simulation platform to the MotionDesk PC is relevant for the currently configured target (xxx.tlc).

Broadcast (use in peer-to-peer network only!) Indicates whether the broadcast mode is used to send the network data packages. If the checkbox is selected, the data packages are sent to all hosts that are connected to the network.

> **Note**
>
> It is highly recommended to use a dedicated local network for your broadcast-based visualization tasks. Do not use the company LAN. The broadcast mode for sending messages produces an enormous amount of network traffic which may seriously impede network operation. If you need a connection to your company LAN on the MotionDesk PCs, use two separated network adapters on them, one for the company LAN and one for your local visualization network.

MicroLabBox Configuration

Relevant for currently configured target (xxx.tlc) (read-only) Indicates whether sending motion data from the simulation platform to the MotionDesk PC is relevant for the currently configured target (xxx.tlc).

Broadcast (use in peer-to-peer network only!) Indicates whether the broadcast mode is used to send the network data packages. If the checkbox is selected, the data packages are sent to all hosts that are connected to the network.

> **Note**
>
> It is highly recommended to use a dedicated local network for your broadcast-based visualization tasks. Do not use the company LAN. The broadcast mode for sending messages produces an enormous amount of network traffic which may seriously impede network operation. If you need a connection to your company LAN on the MotionDesk PCs, use two separated network adapters on them, one for the company LAN and one for your local visualization network.

ConfigurationDesk and VEOS Configuration

Relevant for currently configured target (xxx.tlc) (read-only) Indicates whether sending motion data from the simulation platform to the MotionDesk PC is relevant for the currently configured target (xxx.tlc).

Broadcast (use in peer-to-peer network only!) Indicates whether the broadcast mode is used to send the network data packages. If the checkbox is selected, the data packages are sent to all hosts that are connected to the network.

Note

It is highly recommended to use a dedicated local network for your broadcast-based visualization tasks. Do not use the company LAN. The broadcast mode for sending messages produces an enormous amount of network traffic which may seriously impede network operation. If you need a connection to your company LAN on the MotionDesk PCs, use two separated network adapters on them, one for the company LAN and one for your local visualization network.

Note

Hardware platforms that are supported by ConfigurationDesk include: SCALEXIO, DS6001 and MicroAutoBox III.

MD_Object

Purpose

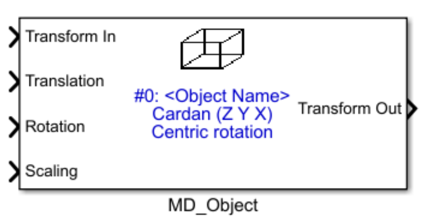
To define a 3-D object and calculate its transformation matrix.

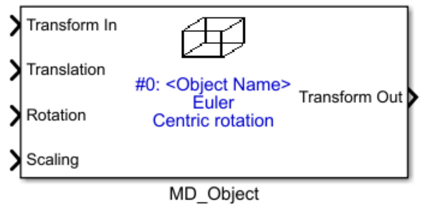
Where to go from here

Information in this section

Block Description (MD_Object)

Illustration





Purpose

To define a 3-D object and calculate its transformation matrix.

Description

This blocks defines a 3-D object for MotionDesk. Additionally, it calculates a new transformation matrix of a 3-D object, starting from the position defined by the Transform In inport. The 3-D object is moved by the Translation vector, rotated by the Rotation vector, and scaled by the Scaling vector. You can select the sequence of translation and rotation and the angle interpretation via block parameters. If the values of translation, rotation and scaling are constant, they can be defined as parameters. In this case they cannot be changed during simulation, but it improves the simulation performance, because the block calculates parts of the transformation matrix only once at the simulation startup.

The transformation matrix of this 3-D object is transferred to MotionDesk via the MD_Communication block which is internally connected to all MD_Object blocks used in the real-time model. An explicit connection in Simulink is therefore not necessary.

Note that the usage of the MD_Communication block, MD_Instrumentation block and MD_Object blocks in the model is limited:

- All used blocks must have the same sample time.
- All used blocks must be in the same atomic subsystem.

I/O characteristics

The following table shows the block inport:

Simulink Input	Range	Simulink Data Type	Meaning
Transform In	4×4 matrix	double	The inport gives the initial values for the transformation matrix. It must be a 4x4 matrix with the data type double. If it is the first block of a kinematic chain, you must connect this port to the MD_Inertial_Coordinate_System block.
Translation	3×1 vector	double	The inport is used to define the object's movement. It must be a vector with 3 elements (x-, y-, z-direction) of data type double. The values must be in meters.
Rotation	3×1 vector	double	The inport is used to define the object's rotation. It must be a vector with 3 elements of data type double. Their values

Simulink Input	Range	Simulink Data Type	Meaning
			must be in degrees. You can define the angles in Euler or Cardan angle format.
Scaling	3×1 vector	double	The inport defines the scaling factors of the object. It must be a vector with 3 elements (x-, y-, z-direction) of data type double. The elements are factors, this means no scaling with `[1, 1, 1]`.

The following table shows the block outport:

Simulink Output	Range	Simulink Data Type	Meaning
Transform Out	4×4 matrix	double	The outport contains the new transformation matrix.

Dialog Page

The dialog settings can be specified on the following dialog page:

- Block Dialog (MD_Object)

Related topics

References

MD_Communication.. 52
MD_Inertial_Coordinate_System.. 66

Block Dialog (MD_Object)

Purpose

To parameterize the MD_Object block.

Dialog settings

Send motion data to MotionDesk Indicates the block functionality. If checked, the block is used as an Object block. If cleared, the block is used as a Transformation block.

Object ID Lets you specify the ID of the 3-D object. The ID must be unique. Due to internal restrictions, you should define IDs starting at 0 and increasing in consecutive order up to a maximum value of 499. The highest ID should not be greater than the number of 3-D objects. Otherwise, too much motion data is transmitted to MotionDesk.

Object name Lets you specify a name for the 3-D object. The maximum length is 20 characters. On some simulation platforms, the name is cut after 20 characters. In a Simulink simulation or when the processor board is connected via MotionDesk Interface Kit, a name longer than 20 characters is allowed. However, the name should be as short as possible because long names can decrease the simulation performance. This name is displayed in MotionDesk when assigning motion data to a movable object in MotionDesk.

Angle interpretation Lets you choose whether the rotation angles are interpreted as Euler or Cardan (roll-pitch-yaw) angles.

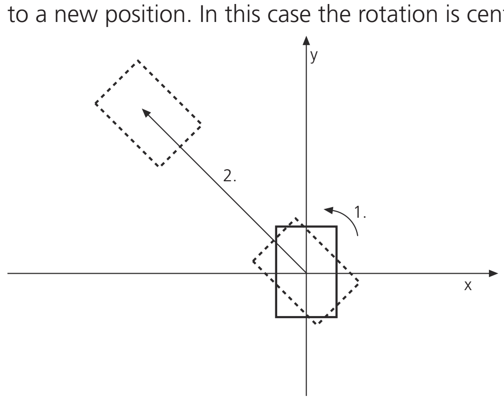
Rotation order Lets you select the rotation order if the angle interpretation is Cardan.

Euler angles use the same axis for both the first and third rotations. For the Euler angle interpretation the rotation order is always Z - Y' - Z''.

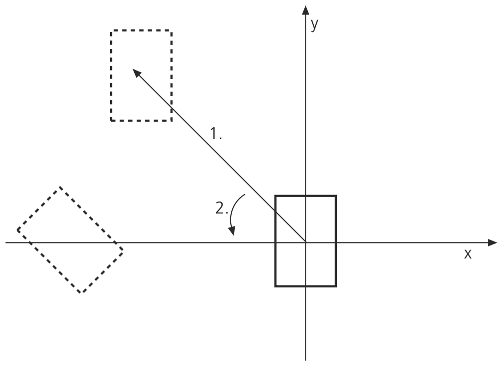
Rotation Order	Description
Z - Y' - X''	Z - Y' - X'': The 3-D object is rotated around its axes in the following manner: 1. Yaw rotation around the z-axis with the Phi Φ angle. 2. Pitch rotation around the y-axis with the Theta θ angle. 3. Roll rotation around the actual x-axis with the Psi ψ angle. The Pitch and Roll steps are performed around newly formed coordinates after the previous rotation completes.
Z - X' - Y''	Z - X' - Y'': The 3-D object is rotated around its axes in the following manner: 1. Yaw rotation around the z-axis with the Phi Φ angle. 2. Roll rotation around the x-axis with the Psi ψ angle. 3. Pitch rotation around the actual y-axis with the Theta θ angle.

Transformation order Lets you choose the sequence of translation and rotation (off-centric or centric rotation). The following two illustrations show the different results when changing the sequence of translation and rotation.

The first illustration shows an object which is first rotated and afterwards moved to a new position. In this case the rotation is centric.



The second illustration shows an object which is first moved to a new position and afterwards rotated. In this case the rotation is off-centric.



Translation as parameter Indicates whether the translation vector is defined by a parameter or inport. If the checkbox is selected, the Translation input is hidden and the Translation edit field is enabled.

Translation Lets you specify the translation vector. The translation vector must have 3 elements (x-, y-, z-direction). Their values must be in meters. The values cannot be changed during the simulation.

Rotation as parameter Indicates whether the rotation angles are defined by a parameter or inport. If the checkbox is selected, the Rotation inport is hidden and the Rotation edit field is enabled.

Rotation Lets you specify the rotation vector. The rotation vector must have 3 elements (Euler or roll-pitch-yaw format, see the Angle interpretation parameter above). Their values must be in degrees. The values cannot be changed during the simulation.

Scaling as parameter Indicates whether the scaling vector is defined by a parameter or inport. If the checkbox is selected, the Scaling inport is hidden and the Scaling edit field is enabled.

Scaling Lets you specify the scaling vector. The scaling vector must have 3 elements (x-, y-, z-direction). The elements are factors, this means no scaling with `[1, 1, 1]`. The values cannot be changed during the simulation.

Related topics

Basics

Cardan Roll, Pitch, and Yaw Angles.......... 16
Euler Angles.......... 15

References

Object Properties (MotionDesk Scene Creation 📖)

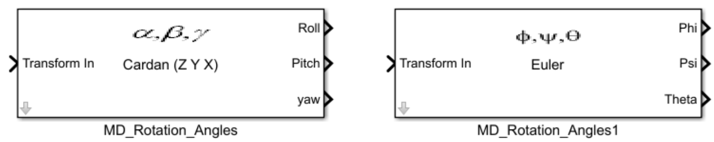
MD_Rotation_Angles

Purpose

To calculate the Euler and roll-pitch-yaw angles from a transformation matrix.

Block Description (MD_Rotation_Angles)

Illustration



Purpose

To calculate the Euler and roll-pitch-yaw angles from a transformation matrix.

Description

A transformation matrix is not suitable for interpreting a rotation for a user. A rotation of a 3-D object can easier be interpreted when it is displayed in 3 separate rotation angles. The angles can be output either as Euler or Cardan angles. You can select the angle system in the block dialog for example:

Cardan system Z - Y' - X'': The 3-D object is rotated around its axes in the following manner:

1. Yaw rotation around the z-axis with the Phi Φ angle.
2. Pitch rotation around the y-axis with the Theta θ angle.
3. Roll rotation around the actual x-axis with the Psi ψ angle.

The Pitch and Roll steps are performed around newly formed coordinates after the previous rotation completes.

> **Note**
>
> The MD_Rotation_Angles block supports only the rotation order in the z-y'-x'' convention.

Euler system Euler angles use the same axis for both the first and third rotations. For the Euler angle interpretation the rotation order is always Z - Y' - Z'':

1. Rotation around the z-axis with the Psi ψ angle
2. Rotation around the y-axis with the Theta Φ angle
3. Rotation around the actual z-axis with the Phi θ angle

Parameters frame

Angle Interpretation Lets you select Cardan or Euler as the angle interpretation system.

I/O characteristics

The following table shows the block inport:

Simulink Input	Range	Simulink Data Type	Meaning
Transform In	4×4 matrix	double	The inport must be a 4x4 transformation matrix.

The following table shows the block outport if *Cardan* is selected:

Simulink Output	Range	Simulink Data Type	Meaning
Roll	-	double	The outport contains the roll angle.
Pitch	-	double	The outport contains the pitch angle.
Yaw	-	double	The outport contains the yaw angle.

The following table shows the block outport if *Euler* is selected:

Simulink Output	Range	Simulink Data Type	Meaning
Phi	-	double	The outport contains the phi angle.
Psi	-	double	The outport contains the psi angle.
Theta	-	double	The outport contains the theta angle.

Related topics

Basics

Cardan Roll, Pitch, and Yaw Angles........ 16
Euler Angles........ 15

MD_Inertial_Coordinate_System

Purpose

To define an inertial coordinate system.

Block Description (MD_Inertial_Coordinate_System)

Illustration

MD_Inertial_Coordinate_System

Purpose

To define an inertial coordinate system.

Description

This block defines the inertial coordinate system for a kinematic chain.

I/O characteristics

The following table shows the block outport:

Simulink Output	Range	Simulink Data Type	Meaning
Transform Out	4×4 matrix	double	The output contains a transformation matrix that can be used as the Transform In inport for an MD_Object block.

Related topics

References

MD_Object.. 60

MD_Car_Kinematic

Purpose

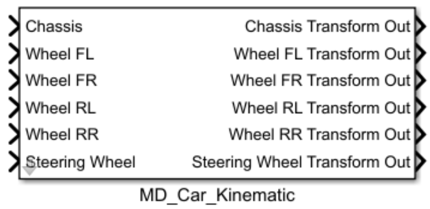
To realize a kinematic chain of a typical car.

Where to go from here

Information in this section

Block Description (MD_Car_Kinematic)

Illustration



Purpose

To realize a kinematic chain of a typical car.

Description

This block is a subsystem that you can use to visualize a typical car with a chassis, 4 wheels and a steering wheel. The distances between the axles and the wheels can be defined as parameters. Because this model is simplified, it has the following characteristics:

- The angles of the rotation of the chassis and the wheels are calculated in the roll-pitch-yaw format.
- The angles of the rotation of the steering wheel are calculated in the Euler format.
- This block does not define movable objects for MotionDesk. For this, the output ports must be connected to MD_Object blocks.

I/O characteristics

The following table shows the block inport:

Simulink Input	Range	Simulink Data Type	Meaning
Chassis	6×1 vector	double	The inport defines the translation and rotation of the car body. It must be a vector with 6 elements. First the translation vector and afterwards the rotation angles in roll-pitch-yaw format.
Wheelxy	6×1 vector	double	The inports define the translation and rotation of the wheels. It must be a vector with 6 elements. First the translation vector and afterwards the rotation angles in roll-pitch-yaw format. The abbreviation xy represents the wheel's position: ▪ FR: front - right ▪ FL: front - left ▪ RL: rear - left ▪ RR: rear - right
Steering Wheel	6x1 vector	double	The inport defines the translation and rotation of the steering wheel. It must be a

Simulink Input	Range	Simulink Data Type	Meaning
			vector with 6 elements. First the relative position of steering wheel to the car's coordinate system and afterwards the rotation angles in Euler format.

The following table shows the block outport:

Simulink Output	Range	Simulink Data Type	Meaning
Chassis Transform Out	4×4 matrix	double	The outports output the transformation matrix of the car body.
Wheelxy Transform Out	4×4 matrix	double	The outports output the transformation matrices of the wheels. The abbreviation xy represents the wheel's position: ▪ FR: front - right ▪ FL: front - left ▪ RL: rear - left ▪ RR: rear - right
Steering Wheel Transform Out	4×4 matrix	double	The outports output the transformation matrix of the steering wheel.

Connect the outports to MD_Object blocks, to define movable objects for MotionDesk.

Dialog page

The dialog settings can be specified on the following dialog page:

- Block Dialog (MD_Car_Kinematic)

Related topics

References

MD_Object.. 60

Block Dialog (MD_Car_Kinematic)

Purpose

To parameterize the MD_Car_Kinematic block.

Dialog settings

Distance between axles Lets you specify the distance between the axles in meters.

Distance between wheels Lets you specify the distance between the wheels of an axle in meters. This value applies to the front and rear axle.

MD_Instrumentation

Purpose

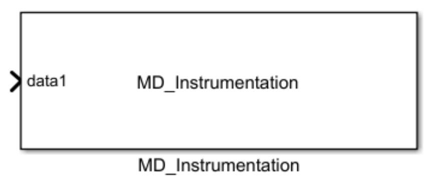
To send scalar data from the simulation model to MotionDesk for visualizing them in instruments.

Where to go from here

Information in this section

Block Description (MD_Instrumentation)

Illustration



Purpose

To send scalar data from the simulation model to MotionDesk for visualizing them in instruments.

Description

The values of the signals which are connected to the block are send to MotionDesk. In MotionDesk they can be connected to instruments, so their values can be monitored in the scene during animation. The number of signals is specified in the block dialog. Up to 300 signals can be sent. You can specify a name, description, unit and the data type for each signal. Only one MD_Instrumentation block must be used in the model.

For basic information on how to use instruments in a scene, refer to Using Instruments in the Scene (MotionDesk Scene Animation 📖).

Note that the usage of the MD_Communication block, MD_Instrumentation block and MD_Object blocks in the model is limited:

- All used blocks must have the same sample time.
- All used blocks must be in the same atomic subsystem.

I/O characteristics

The following table shows the block inport:

Simulink Input	Description
data<*n*>	Lets you connect the Simulink signals to be sent to MotionDesk for the instruments. You can specify the following settings for the inports on the block dialog: ▪ Number and order of inports ▪ Name, description, unit, and data type of each variable

Dialog page

The dialog settings can be specified on the following dialog page:

- Block Dialog (MD_Instrumentation)

Block Dialog (MD_Instrumentation)

Purpose

To specify the settings for sending signal values to MotionDesk instruments.

Dialog settings

Instrument data ports Lets you add, delete and sort data ports.

Button	Description
+	Lets you add a data port below the selected data port
-	Lets you delete the selected data port.
↑	Lets you move the selected data port upwards
↓	Lets you move the selected data port downwards

Name Lets you specify a name for the data port. The name is the signal name in MotionDesk. All instrument data streams must have unique names. The maximum length is 20 characters. On some simulation platforms, the name is cut after 20 characters. In a Simulink simulation or when the processor board is connected via MotionDesk Multi-PC Interface Kit, a name longer than 20 characters is allowed. However, the name should be as short as possible because long names can decrease the simulation performance.

Description Lets you specify a description for the signal.

Unit Lets you specify a string that is the unit for the signal, for example, "km/h". The unit is displayed in the MotionDesk instrument to which the signal is connected.

Type Lets you select a data type for the signal.

MD_Set_Sample_Rate and MD_Subsystem

Purpose

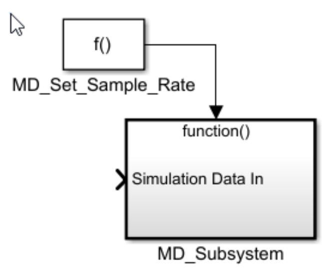
To create a subsystem with MotionDesk Blockset blocks.

Where to go from here

Information in this section

MD_Set_Sample_Rate and MD_Subsystem

Illustration



Purpose

To create a subsystem with MotionDesk Blockset blocks.

Description

Generally the step size of the visualization (frame rate) is greater than the step size of the simulation. It is therefore not necessary to calculate motion data at every simulation step. It is recommended to use the MD_Subsystem template to trigger the MotionDesk Blockset blocks in time steps that are sufficient for your application. The sample time must be adapted to the use scenario, for example, for a man-in-the-loop application you should set a low sample time to get a low latency time.

The 3-D subsystem template consists of an MD_Set_Sample_Rate block and an MD_Subsystem block. For a description of these blocks, see below.

Block Description (MD_Set_Sample_Rate)

Block

Purpose

To trigger the 3-D subsystem.

Description

This block generates a trigger signal for the 3-D subsystem. The interval between the trigger signals is defined by the block parameters and depends on the usage scenario. Typical values are, for example, 10 ms for a man-in-the-loop scenario with not more than 30 movable objects or 20 ms for an automatic maneuver scenario with up to 150 movable objects.

> **Note**
>
> Using very short trigger times (for example, 1 ms) can cause buffer problems, because the shorter the trigger time is, the faster the buffer is filled.
> Lossless data acquisition requires a trigger of at least 10 ms.

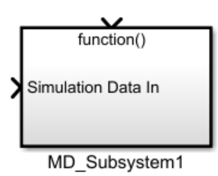
Parameters frame

Sample time Lets you specify the sample time.

Number of iterations Lets you specify the number of iterations.

Block Description (MD_Subsystem)

Block



Purpose

To calculate the motion data in defined time intervals.

Description

The MD_Subsystem block must contain all MotionDesk Blockset blocks that are necessary to calculate motion data.

I/O characteristics

The following table shows the block inport:

Simulink Input	Range	Simulink Data Type	Meaning
Simulation Data In	-	double	The inport is the connection for the simulation data which are needed to calculate the position and orientation (kinematic) of the 3-D objects for the animation. Connect all variables to be visualized to this port.

MotionDesk Service Library

Introduction

MotionDesk software contains the MotionDesk Service Library for adapting a real-time application to a MotionDesk animation. Use this library as an interface between the real-time application and MotionDesk.

Where to go from here

Information in this section

Data Types for MotionDesk Services

Introduction

The following data types are defined for the MotionDesk services.

Data types

md_error_type Data type for returned error codes

```
typedef int    md_error_type;
```

md_state_type Data type for defining the MotionDesk simulation state

```
typedef enum
{
   RM_DSP_STOP = 0,
   RM_DSP_PAUSE,
   RM_DSP_RUN
} md_state_type;
```

md_homog_member_data_type 4 x 4 matrix data type for holding homogeneous member data, combining translation and transformation data

```
typedef double md_homog_member_data_type[4][4];
```

md_p_homog_member_data_type Pointer to a homogeneous member data matrix

```
typedef md_homog_member_data_type* md_p_homog_member_data_type;
```

RM_HOMOG_MEMBER_DATA_PTR Pointer to a homogeneous member data structure:

```
typedef RM_HOMOG_MEMBER_DATA*  RM_HOMOG_MEMBER_DATA;
```

md_init_interface

Syntax

```
md_error_type md_init_interface(
   UInt32  number_of_members)
```

Include file

`mdserv.h`

Purpose

To initialize the interface for the MotionDesk service functions.

Description

The `md_init_interface` function initializes the MotionDesk service module. It must be called once before all other MotionDesk service functions. If this function is not called, the `md_set_member_name` function returns the error code `MD_NOT_INITIALIZED`. The other service functions, which are usually used under real-time conditions, do not perform this check for performance reasons. The `md_init_interface` function allocates memory for the member data and member names structures and initializes them. The MotionDesk simulation state, which indicates the state of the simulation, is set to `MD_SIM_STOP`. The simulation state can be changed by the `md_set_sim_state` function.

Parameters

number_of_members Number of data members for which memory space will be allocated

Return value

Error code. The following values are predefined:

Predefined Symbols	Meaning
MD_NO_ERROR	No error
MD_OUT_OF_MEMORY	Dynamic memory missing or too small

Example

The following example shows how to use the function:

```
md_error_type errcode
UInt32 num_members;
...
/* initialize the MotionDesk service for 12 data members */
num_members = 12;
errcode = md_init_interface(num_members);
```

Related topics

Basics

Working with the MotionDesk Service Library.. 46

References

Data Types for MotionDesk Services.. 75
md_set_sim_state... 80

md_set_member_name

Syntax

```
md_error_type md_set_member_name(
   UInt32 member_id,
   char member_name[])
```

Include file

`mdserv.h`

Purpose

To set a member name.

Description

The `md_set_member_name` function is optional and can be called only after `md_init_interface` and before the first call of an interrupt service routine. The function defines names for members, so that they can be identified.

Parameters

member_id Member index within the range 0 … (number of elements – 1). The number of elements is specified with the `md_init_interface` function.

member_name Name of the member. The maximum length is 20 characters. If member_name has more characters, only the first 20 characters are used.

Return value

Error code. The following values are predefined:

Predefined Symbols	Meaning
MD_NO_ERROR	No error
MD_OUT_OF_RANGE	The specified index is out of range.
MD_NOT_INITIALIZED	The call of the `md_init_interface` function is missing or has failed.

Example

The following example shows how to use the function:

```
md_error_type errcode;
...
/* assign name ’steering wheel’ to member with index 0 */
errcode = md_set_member_name(0, "steering_wheel");
...
```

Related topics

Basics

Working with the MotionDesk Service Library.......... 46

References

Data Types for MotionDesk Services.......... 75
md_init_interface.......... 76

md_set_member_position

Syntax

```
void md_set_member_position(
   unsigned int member_id,
   RM_HOMOG_MEMBER_DATA_PTR homog_data_ptr)
```

Include file

`mdserv.h`

Purpose

To set the position of a single data member.

Description

Use this function to set the position of a single data member with a homogeneous data matrix.

You must initialize the interface with the `md_init_interface` function before.

Parameters

member_id Member index within the range 0 … (number of elements – 1). The number of elements is specified with the `md_init_interface` function.

homog_data_ptr Pointer to a homogeneous data structure

Example

The following example shows how to use the function:

```
...
md_p_homog_member_data_type member0_homog_ptr;
...
/* calculate the homogeneous data matrix of member 0 */
...
/* set the position of member 0 */
md_set_member_position(0, member0_homog_ptr);
...
```

Related topics

References

md_service

Syntax

```
void md_service(
   double tsim)
```

Include file

`mdserv.h`

Purpose

To transfer all data from the real-time application to MotionDesk.

Description

The function transfers all data from the internal model data buffer to the member data buffer, which is used for the data transfer to MotionDesk. Values in TI floating-point format are implicitly converted to the IEEE format.

This function has to be called in the model foreground after all member data is calculated and the member positions are set using the `md_set_member_position` function.

You must initialize the interface with the `md_init_interface` function beforehand.

Parameters

tsim Simulation time for the current set of data members

Example

The following example shows how to use the function:

```
...
double tsim;
...
void foreground_isr()
{
   /* calculate the model parameters */
   ...
   /* set the new member positions */
   ...
   md_service(tsim);
}
```

Related topics

References

md_init_interface 76
md_set_member_position 78

md_set_sim_state

Syntax

```
void md_set_sim_state(
   UInt32 sim_state)
```

Include file

`mdserv.h`

Purpose

To set the simulation running state.

Description

Every time the simulation state changes, it must be set by using this function. MotionDesk reads the simulation state to decide whether it has to display a new frame for the delivered member data or not.

You must initialize the interface with the `md_init_interface` function before.

Parameters

sim_state Current simulation state. The following symbols are predefined:

Predefined Symbol	Description
`MD_SIM_STOP`	Simulation state stop
`MD_SIM_PAUSE`	Simulation state pause
`MD_SIM_RUN`	Simulation state run

Example

The following example shows how to use the function:

```
...
md_state_type sim_state;
...
/* set the MotionDesk simulation state to run */
sim_state = MD_SIM_RUN;
md_set_sim_state(sim_state);
...
```

Related topics

References

md_set_maneuver_support

Syntax

```
void md_set_maneuver_support (RM_MANEUVER_SUPPORT maneuver_support)
```

Include file

`mdserv.h`

Purpose

To set whether recording synchronization is supported or not.

Description

`md_init_interface` uses `RM_MANEUVER_UNSUPPORTED` as the default value. If you want to use recording synchronization, you have to enable its support with `RM_MANEUVER_SUPPORTED`. Call this function after `md_init_interface`.

Recording synchronization ensures that the simulation and the reference tracks run synchronously by resetting the buffers and starting the simulation and the tracks simultaneously. Refer to Basics of the Multi-Track Mode (MotionDesk Scene Animation 📖).

`md_set_maneuver_support` corresponds to Support recording synchronization option in the MD_Communication block of the MotionDesk Blockset. Refer to Block Description (MD_Communication) on page 53.

Parameter

maneuver_support Support of recording synchronization. The following symbols are predefined:

Predefined Symbol	Description
`RM_MANEUVER_UNSUPPORTED`	Recording synchronization is not supported
`RM_MANEUVER_SUPPORTED`	Recording synchronization is supported

Return value

None

Related topics

References

Data Types for MotionDesk Services 75
md_init_interface 76

md_set_maneuver_start_time

Syntax

```
void md_set_maneuver_start_time (double maneuver_start_time)
```

Include file

`mdserv.h`

Purpose

To set the recording start time.

Description

When time stamps for motion data are calculated, the recording start time is subtracted from the actual simulation time. The time stamps of the motion data start at zero at `maneuver_start_time`.

You should call `md_set_maneuver_start_time()` in the following cases:

- When you start a new maneuver.
- When an event occurs for which you want to start a new recording, for example, if a vehicle reaches 60 km/h.

Parameter	**maneuver_start_time**	The start time of a recording.

Return value None

Related topics

References

md_close_interface

Syntax

```
void md_close_interface()
```

Include file `mdserv.h`

Purpose To terminate the service module.

Description The function sets the MotionDesk interface to an initial state and frees all memory resources allocated by the service module.

Parameters None

Example The following example shows how to use the function:

```
...
/* terminate the service module */
md_close_interface();
...
```

Related topics

References

Troubleshooting (Real-Time Application)

Introduction

MotionDesk provides a diagnostic tool that can display the numerical values of motion data for debugging purposes.

Where to go from here

Information in this section

Basics of Motion Diagnostics

Introduction

MotionDesk provides a diagnostic tool that can display the numerical values of motion data for debugging purposes.

Motion Diagnostics

Generally, the simulated values are not visible in MotionDesk. However, if your real-time application is erroneous, you might want to get the numerical values for debugging purposes. MotionDesk provides a diagnostic tool for this, called Motion Diagnostics.

Motion Diagnostics displays the position and orientation of a selected object in numerical values. The position and orientation are given by the rotation matrix and translation vector. Because the orientation is hard to recognize when only the rotation matrix is given, the rotating angles are additionally displayed in Euler or roll-pitch-yaw format.

How to Use Motion Diagnostics

Objective

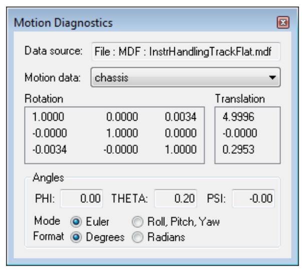
Motion Diagnostics can run in an animation for diagnostics purposes. The values are only updated when an animation is running. The Motion Diagnostics pane displays a transformation matrix and rotation angles of a simulated object.

Method

To use Motion Diagnostics

1 In the View ribbon, click Controlbars – Switch Controlbars and select Motion Diagnostics.

MotionDesk opens the Motion Diagnostics pane. The current selected data source is displayed in the Data source field.



2 In the Motion data field, select the object whose values need to be displayed.

3 Select the mode of the angle values.

4 Select the format of the angle values.

Result

When an animation is running, the current transformation and rotation matrices and rotation angles of the object are displayed.

Related topics

Basics

Cardan Roll, Pitch, and Yaw Angles........16
Euler Angles........15
Homogeneous Transformation........14

References

Motion Diagnostics (MotionDesk Scene Animation 📖)

Limitations

Limitations when Modeling Using the MotionDesk Blockset

Introduction	There are a number of limitations for implementing models with the MotionDesk Blockset.
Using MotionDesk Blockset blocks	Limitations for using the MD_Communication block and the MD_Object blocks in a model: ▪ All blocks must have the same sample time. ▪ All blocks must be in the same atomic subsystem.
Excluding blocks from simulation	Commenting out or commenting through MotionDesk Blockset blocks in your model to exclude them from simulation is not supported.
Limitations applying to referenced models	MotionDesk Blockset blocks are not supported in referenced models.
Row-major code generation	The MotionDesk Blockset does not support row-major code generation that is introduced by MATLAB 2018b.
Restricted support of Variant Subsystem block	There must be no MotionDesk Blockset blocks in a Variant Subsystem or any of its subsystems, whether activated or deactivated.
Using MotionDesk Blockset blocks in an MP system	If your simulation model runs on a multiprocessor system, you must use the MotionDesk Blockset blocks only in one submodel.

Using MotionDesk Blockset blocks in Simulink subsystems	Using MotionDesk Blockset blocks in the following types of Simulink subsystems is not supported: ▪ Simulink Function subsystem ▪ Simulink subsystems with read/write permissions set to `NoReadOrWrite`
Maximum values	**Motion data** The maximum number of MD_Object blocks is 500. **Signal data** The maximum number of signals which can be transfered to MotionDesk is 300.
Data assignment	The assignment of objects to motion data is based on an index. If you record MDF files and change the order of IDs in your simulation model, the motion data must be reassigned to the movable objects.
Rapid Accelerator mode	The blocks of the MotionDesk Blockset can be used in a Simulink simulation in the Rapid Accelerator mode. The following limitations apply: ▪ You need a C/C++ compiler to build MEX files. For more information on the supported software and versions, refer to C/C++ compiler for building MEX files (nonTargetLink) in Required C and C++ Compilers (Installing dSPACE Software 📖). ▪ Only one simulation must be executed at the same time, regardless of which simulation mode is used.
Task configuration in ConfigurationDesk for SCALEXIO systems	The task configuration in ConfigurationDesk is limited when the behavior model contains MotionDesk Blockset blocks. In the model, the following options for the jitter and latency optimization for a task within the same application process are not supported: ▪ `No jitter, low latency` ▪ `Low jitter, low latency`
Parallel simulation	You can simulate a model parallel using different constraints, for example, different parameter values or initial values. The MotionDesk Blockset does not support parallel simulation, because MotionDesk requires a continuous data stream of motion data.

Related topics

Basics

Limitations (RTI and RTI-MP Implementation Guide 📖)

References

Required C and C++ Compilers (Installing dSPACE Software 📖)

Numerics