Open Source Simulink Block for Vicon System

Augustin Manecy
July 21, 2014

Contents

1 Requirements 2

2 Integration of the Vicon tools in Simulink 2
  2.1 Using the RT-MaG toolbox 3
  2.2 Without the RT-MaG toolbox 3

3 How to use it? 4
  3.1 Prepare your folder 4
  3.2 Configure the Vicon Simulink block 5
    3.2.1 The parameters 5
    3.2.2 The outputs 6
  3.3 Run Simulation mode 6
  3.4 Real-Time mode 7

4 How to Tune the data received from the Vicon block 9
1 Requirements

To be able to use this Simulink block, you need:

- A Vicon System,
- Tracker and/or Nexus,
- The Vicon SDK for Windows,
- The archive ViconSimulinkBlock_v1.0.zip.

Remark 1: If you want to achieve accurate real-time performances, it is really recommended to use a real-time windows target to execute the simulink model with the Vicon block. e.g.: you can use Quarc® of Quanser or Real-Time Windows Target™ of Mathworks®.

2 Integration of the Vicon tools in Simulink

1. First you need to install the Vicon SDK for Windows if it is not already done. Then you have to add the path of the Vicon C++ SDK to your WINDOWS path environment variables (figure 1.a).

2. Add the path of the Vicon SDK to the MATLAB path (figure 1.b).

![Figure 1: a) Add the path of the Vicon SDK installation folder (by default: C:\Program Files\Vicon\DataStream SDK\Win64\CPP) to your Windows path. We recommend to you to copy the previous environment variables in a text file to avoid wrong manipulations. The just restart your computer to make the path modification active. b) Add the path of the Vicon SDK installation folder to the Matlab path and save the new path.](image-url)
2.1 Using the RT-MaG toolbox

If you have the RT-MaG toolbox installed on your computer you can find the Simulink Vicon block in the Simulink Library Browser. Take a look at the figure 2.

![The Vicon Reception Block](image1.png)

Figure 2: The HOST library of the RT-MaG toolbox with the Vicon reception block.

2.2 Without the RT-MaG toolbox

If you have not the RT-MaG toolbox installed on your computer, you just need to add the directory `ViconBlockResources` to your Matlab path. See figure 3.

**Remark 2:** You have to run Matlab with the administrator privileges if you want to change definitively the matlab path.

![Add the folder](image2.png)

Figure 3: Add the folder `ViconBlockResources` to your Matlab path with administrator privileges.
3 How to use it?

3.1 Prepare your folder

Step 1: As Tracker and Nexus are licensed softwares, you need to add your own files to your working directories in Matlab. i.e., you have to copy the following files in the directory of your Simulink model:

- The directory *Microsoft.VC90.CRT*,
- *boost_thread-vc90-mt-1_43.dll*
- Client.h
- *ViconDataStreamSDK_CPP.dll*
- *ViconDataStreamSDK_CPP.lib*

**Remark 3:** You can find these files in the following directory:
C:\Program Files\Vicon\DataStream SDK\Win64\CPP

Step 2: Next you have to add to the directory the C++ files of the S-Function which allow to communicate via Simulink to the Vicon System. i.e., You have to copy the following files in the same directory of your Simulink model:

- SfunCPP_RecoverVicon_SDK.cpp
- SfunCPP_RecoverVicon_SDK.h

**Remark 4:** You can find these files in the `RealTime_Example` or in the `Simulation_Example` directory of the archive `ViconSimulinkBlock_v1.0.zip`

Step 3: Now you have to generate the *.mexw64* file (corresponding to `SfunCPP_RecoverVicon_SDK.cpp`) which will be used by Simulink to communicate with the Vicon System. To do this, you just need to type in your matlab command window:

```
>> mex SfunCPP\RecoverVicon\SDK.cpp
```

**Remark 5:** You need to check that a compiler is linked to Matlab, you can check this by typing `mex -setup` in your matlab command window as described by figure 4. You can for example use Visual Studio Express Edition.

![Figure 4: Configure your compiler in MATLAB.](Image)
3.2 Configure the Vicon Simulink block

The Vicon block system allows to receive data about segments, labelled markers and unlabelled markers available on the Vicon system. To simplify the use, the Vicon Simulink block provides a user-friendly mask allowing to choose the different data to recover.

3.2.1 The parameters

- **Description:** Active a "ServerPush" on the remote computer running Tracker or Nexus and receive the data Segments, labelled markers and unlabelled markers.

- **Connection Parameters:**
  - IP address: the IP address (IPv4) of the remote computer running Tracker or Nexus. e.g.: 139.124.59.131
  - Port: the port of the connection (default 801). e.g.: 1024

- **List of Segments:**
  - Manual add/remove: You can manually add segment by typing its name (e.g.: SubjectName.SegmentName) in the panel 2 (figure 5) and clicking on the "+" button. You can also remove a segment by selecting it in the panel 1 and clicking on the button ".-".
  - Choose Vicon Segment: You can automatically ask to the Vicon computer to give you a list of available segments using the "..." button (it opens the Vicon Choose Segment panel). In the Vicon Choose Segment panel, you can add/remove a segment thanks to the ">>" and "<<" button and apply change to up to date the List of Segments.

- **List of Markers:**
  - Manual add/remove: You can manually add segment by typing its name (e.g.: SubjectName.MarkerName) in the panel 4 (figure 5) and clicking on the "+" button. You can also remove a marker by selecting it in the panel 3 and clicking on the button ".-".
  - Choose Vicon Marker: You can automatically ask to the Vicon computer to give you a list of available markers using the "..." button (it opens the Vicon Choose Marker panel). In the Vicon Choose Marker panel, you can add/remove a marker thanks to the ">>" and "<<" button and apply change to up to date the List of Markers.

- **Unlabelled Marker Index:** You can recover unlabelled marker (marker which does not belong to a segment) by typing their index. e.g.: [1 4 8], or [1;4;8], or [1], or []
• Sample Time [s]: The sample time of the block (have to be greater or equal to the one of the Vicon system). e.g.: 0.002

3.2.2 The outputs
The outputs of the Vicon Block are the following:

• NewStream?: Is there a new frame for this step time?
  Default dimension: [1 × 1]
  Type: Double
  Value: 1, if a new stream was received, 0 else.

• Frame Number: The number of the frame which is currently used,
  Default dimension: [1 × 1],
  Type: Double
  Value: 1 to 2^{64}

• Latency [s]: The latency in seconds estimated by the Vicon system (does not include the network latency between the computer running the Simulink model and the computer running Tracker or Nexus).
  Default dimension: [1 × 1]
  Type: Double
  Value: latency in seconds (you can expect less than 5 [ms]),

• SEG: SubjectName.SegmentName: The data for the segment SubjectName.SegmentName. There is as many output of this kind as the number of segment which appear in the "List of Segments".
  Default dimension: [7 × 1]
  Type: Double
  Value (by default):
  [1 : 3], position X, Y and Z in millimetre,
  [4 : 6], Euler angles in radians according to the \((X, Y', Z'')\) intrinsic formalism,
  [7], Segment Occluded? 0 if the Segment is seen by the Vicon 1 else.
  **Remark 6:** If you want to customize the data received for a segment, you can take a look at section 4.

• MRK: SubjectName.MarkerName: The data for the labelled marker SubjectName.MarkerName. There is as many output of this kind as the number of marker which appear in the "List of Markers".
  Default dimension: [3 × 1]
  Type: Double
  Value (by default):
  [1 : 3], position X, Y and Z in millimetre,
  **Remark 7:** If you want to customize the data received for a labelled marker, you can take a look at section 4.

• Unlabeled Marker n°X: The data for the unlabelled marker number X. There is as many output of this kind as the number of marker which appear in the vector "Unlabelled Marker Index".
  Default dimension: [3 × 1]
  Type: Double
  Value (by default):
  [1 : 3], position X, Y and Z in millimetre,
  **Remark 8:** If you want to customize the data received for an unlabelled marker, you can take a look at section 4.

3.3 Run Simulation mode
In this mode, the real-time is not ensure, but if your Simulink model doesn’t need a lot of computational resources you can easily achieve real-time performance until 500Hz.

You just need to create a new Simulink model and put the Vicon block on your model, and follow the next instructions:

**Step 1:** Configure your Vicon block:
• Configure the IP of the Vicon server computer (the computer running Tracker or Nexus),
• Configure the Port of the server (by default the server is running on the port 8001),
Select the segment(s) you desire to recover,
Select the labelled marker(s) you desire to recover,
Select the unlabelled marker(s) you desire to recover,
Choose the SAME sample time of your Vicon System. If Tracker (or Nexus) is running at 500Hz, choose 0.002s as SampleTime.

Step 2: Configure the model:

- In Configuration Parameters/Solver, we recommend you to configure the solver like this: Fixed-Step, ode1 (Euler) and the Fixed-step size at the same sample time of your Vicon Block.
- In Configuration Parameters/Code Generation use the System target file ert.tlc,
- In Configuration Parameters/Code Generation/Custom Code: in the Panel Libraries just add .\ViconDataStreamSDK\CPP.lib,
- In File/Model Properties... under the CallBacks panel and in the InitFcn enter mex SfunCPP_RecoverVicon_SDK.cpp.

An example of the Simulation mode is available in the folder Simulation_Example: This simple model (see figure 6) shows you how to recover:

- A segment: A segment is a part of an object defined on Nexus or Tracker. A segment is defined as "ObjectName.SegmentName". In Tracker each object is composed by one and only one segment, so it is defined as "SegmentName.SegmentName".
- A labelled marker: A marker is one element of a segment and is labelled via Nexus or Tracker.
- An unlabelled marker: An unlabelled marker is a marker seen by the Vicon system but which does not belong to a segment. An unlabelled marker is represented by a number.

3.4 Real-Time mode

If you want to use the block in a real-time context, you have to use a software as Quarc® of Quanser or Real-Time Windows Target® of Mathworks®.

Here we explain how to configure your Simulink model and run it in real-time with Quarc® of Quanser.

Step 1: Configure your Vicon block:
- Configure the IP of the Vicon server computer (the computer running Tracker or Nexus),
- Configure the Port of the server (by default the server is running on the port 8001),
- Select the segment(s) you desire to recover,
- Select the labelled marker(s) you desire to recover,
- Select the unlabelled marker(s) you desire to recover,
- Choose a GREATER or EQUAL sample time of your Vicon System. If Tracker (or Nexus) is running at 500Hz, choose 0.002s or more.

**Step 2:** Configure the model:
- Choose external as Simulation mode (See figure 7),
- In Configuration Parameters/Solver, we recommend you to configure the solver like this: Fixed-Step, ode1 (Euler) and the Fixed-step size at the same sample time of your Vicon Block.
- In Configuration Parameters/Code Generation use the System target file ert.tlc,
- In Configuration Parameters/Code Generation/Custom Code: in the Panel Libraries just add \ViconDataStreamSDK\CPP.lib,
- In File/Model Properties... under the CallBacks panel and in the InitFcn enter mex SfunCPP_RecoverVicon_SDK.cpp.

Figure 7: Select the external Model, the connect to target and then run the real-time simulation by clicking on the "play" button.

An example of the real-time mode is available in the folder **RealTime_Example**: This simple model (see figure 6) shows you how to recover:

- **A segment**: A segment is a part of an object defined on Nexus or Tracker. A segment is defined as "ObjectName.SegmentName". In Tracker each object is composed by one and only one segment, so it is defined as "SegmentName.SegmentName".

- **A labelled marker**: A marker is one element of a segment and is labelled via Nexus or Tracker.

- **An unlabelled marker**: An unlabelled marker is a marker seen by the Vicon system but which does not belong to a segment. An unlabelled marker is represented by a number.
4 How to Tune the data received from the Vicon block

If you want you can tune the data receive for each object (Euler angle, Quaternion, Rotation matrix, Global translation...) To know all the data which are available please take a look to the documentation of the Vicon SDK (Vicon DataStream SDK Manual.pdf located in C:\Program Files\Vicon\DataStream SDK\Documentation.

Then it is possible to choose the data by modifying the C++ S-function SfunCPP_RecoverVicon_SDK.cpp, SfunCPP_RecoverVicon_SDK.h and ViconOutputProcessing.m. Just follow the next instructions:

1. In SfunCPP_RecoverVicon_SDK.cpp, you have to modify the lines following the line 357 as your convenience by calling the SDK function you need. i.e., you just need to modify the mdlOuputs function by following the model given in the original file from the line 357 to 372.

2. If the size of the output has changed you have to change the value of the SEGMENT_LENGTH define, available at the line 210 of SfunCPP_RecoverVicon_SDK.h.

3. Change also the dimension of the variable SegmentLength in the file ViconOutputProcessing.m to be the same as the previous define.

Remark 9: You can also change the size and the type of data for the labelled markers and the unlabelled markers by following the same procedure.