

Automatic Gait Events

Overview

Gait events describe key moments in the walking or running cycle - such as heel-strikes and toe-offs. If [Force Platform assignments](#) exist for a motion trial, the **Automatic_Gait_Events** command automatically detects these gait events and places an **event label** at the frames where the subject steps ON and OFF the surface of the platform.

Before running this command, the model must have segments created so that the program can recognize which is left and right. These segments are not limited to feet, they simply describe the part of body which makes contact with the force platform.

Contact with the floor for which there are *no force platforms* can be determined using **Temporal Proximity Rule (TPR)**, this will be seen in the dialog box for this command. No signals need to be provided for the TPR because Visual3D uses the kinematics of the segment making contact with the force platform.

Pipeline Command

This pipeline command was updated in version 5 so that users no longer need so specify the direction of gravity, as Visual3D can determine this internally. Both versions of the command are listed below.

Version 5+	<pre>Automatic_Gait_Events ! /FRAME_WINDOW=8 ! /USE_TPR=TRUE ! /TPR_EVENT_INSTANCE=1 ;</pre>
Older Versions	<pre>Automatic_Gait_Events ! /SELECT_X=FALSE ! /SELECT_Y=FALSE ! /SELECT_Z=TRUE ! /FRAME_WINDOW=8 ! /USE_TPR=TRUE ;</pre>

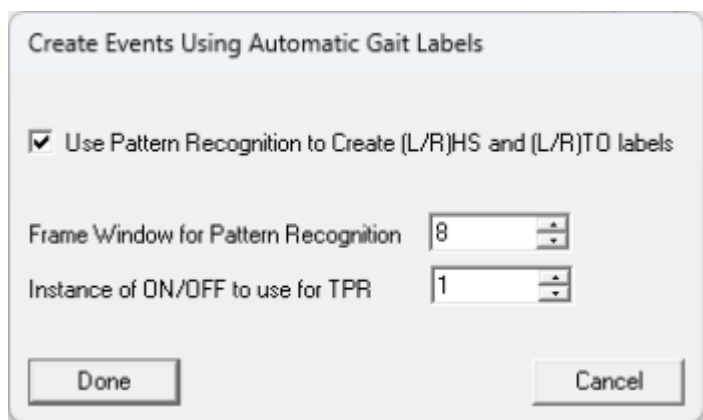
Command Parameters

The following table shows the command parameters for **Version 5+**.

Automatic_Gait_Events	Parameter Description
! /FRAME_WINDOW = 8	The frame window used for pattern recognition (default = 8)
! /USE_TPR=TRUE	Enables TPR Algorithm is used to find the heelstrike (HS) and toeff (TO).
! /TPR_EVENT_INSTANCE=1	Instance used of detected event.

Dialog

The dialog for the command can be accessed by double-clicking it the left mouse button while placed in the pipeline.



- **Use Pattern Recognition to Create (L/R)HS and (L/R)TO Labels:** applies pattern recognition to identify gait cycle events.
- **Frame Window for Pattern Recognition:** Number of frames used for pattern recognition
- **Instance of ON/OFF to use for TPR:** Determines which instance of an ON/OFF transition is used for event detection with TPR.

Examples

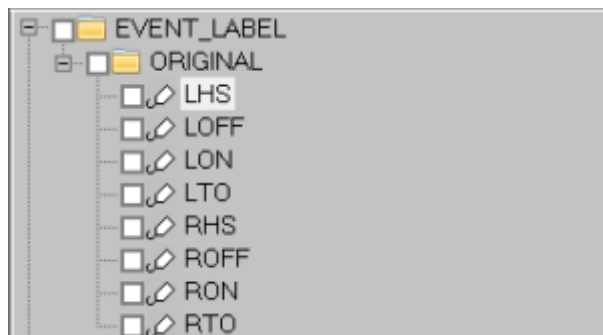
The following examples will show different methods in which `Automatic_Gait_Events` can be used alongside other commands for user applications.

Example 1

The first example shows the functionality of the **Automatic_Gait_Events** command. After loading in your workspace containing a static and dynamic trial (with force assignments), you can simply add the command into the pipeline, which will look as follows:

```
Automatic_Gait_Events
! /FRAME_WINDOW=8
! /USE_TPR=TRUE
! /TPR_EVENT_INSTANCE=1
;
```

After executing the pipeline, you will notice a new folder in your Signals and Events tab called **EVENT_LABEL**, within it is all of the gait events which the program automatically detected and set.



Example 2

In the next example, the **Automatic_Gait_Events** command will be used alongside others in a pipeline which allows the user to do the following starting from an empty project workspace:

- Open the user-selected workspace file. ([File_Open](#))
- **Automatically detect gait events.**
- Iterate through and highlight LHS, LTO, and BTWN event labels. ([For_Each](#)) → ([Highlight_Event_Label](#))
- Switch to Signals and Events tab, and graph X-component of Force Platform signal (events will be shown on this graph). ([Switch_Between_Tabs](#)) → ([Interactive_Graph_Signals](#))

The pipeline will look as follows:

```
File_Open
/FILE_NAME=
! /FILE_PATH=
! /SEARCH_SUBFOLDERS=FALSE
! /SUFFIX=
! /SET_PROMPT=File_Open
! /ON_FILE_NOT_FOUND=PROMPT
! /FILE_TYPES_ON_PROMPT=
;

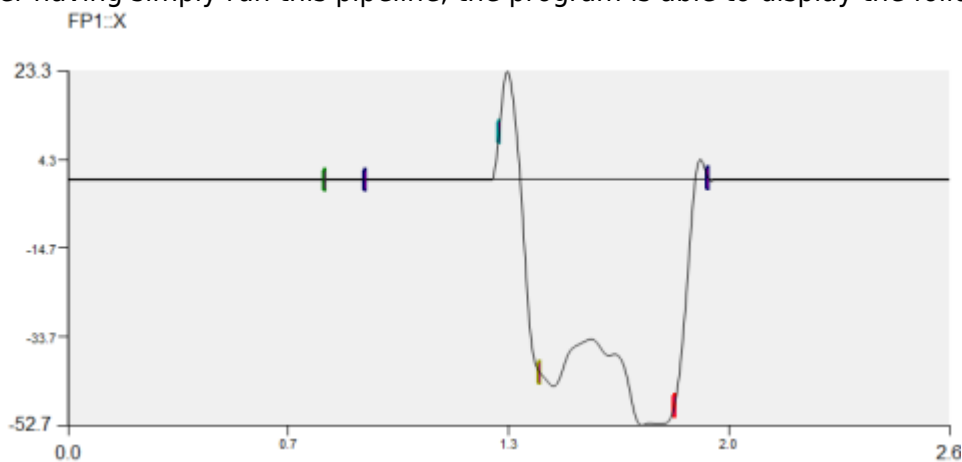
Automatic_Gait_Events
! /FRAME_WINDOW=8
! /USE_TPR=TRUE
! /TPR_EVENT_INSTANCE=1
;

For_Each
/ITERATION_PARAMETER_NAME=EVENT
! /ITERATION_PARAMETER_COUNT_NAME=
/ITEMS= LHS+LTO+LOFF+LON+RHS+RTO+ROFF+RON
;

Highlight_Event_Label
/EVENT_LABEL= ::EVENT
! /REMOVE_EXISTING_HIGHLIGHTS=FALSE
```

```
;  
  
End_For_Each  
/ITERATION_PARAMETER_NAME=EVENT  
;  
  
Switch_Between_Tabs  
/SHOW_SIGNALS_AND_EVENTS_TAB=TRUE  
! /SHOW_WORKSPACE_TAB=FALSE  
! /SHOW_MODELS_TAB=FALSE  
;  
  
Interactive_Graph_Signals  
/SIGNAL_TYPES=FORCE  
/SIGNAL_FOLDER=ORIGINAL  
/SIGNAL_NAMES=FP1  
/SIGNAL_COMPONENTS=X  
/GRAPH_INDEX=1  
/GRAPH_SUBINDEX=1  
! /ZOOM_START_TIME=  
! /ZOOM_END_TIME=  
/REPLACE_CURRENT=TRUE
```

After having simply run this pipeline, the program is able to display the following graph:



From:
<https://has-motion.com/wiki/> - Wiki Documentation Site

Permanent link:
https://has-motion.com/wiki/doku.php?id=visual3d:documentation:pipeline:event_commands:automatic_gait_events&rev=1739981338

Last update: 2025/02/19 16:08

