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 the left from the right. These segments are not limited to feet, they simply describe the part of body which makes contact with the force platform. Contacts with the floor for which there are *no force platforms* can then be determined using **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.

The following is an overall outline of how to use this command. As a reminder, the **Automatic_Gait_Events** pipeline command can be used to create gait events if force platforms exist in a dynamic trial and ON/OFF events indicate kinetic cycles and HS/TO indicate kinematic cycles.

1. A static trial must be loaded into the workspace and a minimum of two segments must be created (for standard gait trials, this means a left/right foot segment must be created).
2. If a force platform assignment exists in a dynamic trial, ON/OFF events will be created when Force Assignments exist. A Force Assignment exists IF:
    1. A segment's center of mass (ANY segment) falls within the specified distance between center of force pressure to segment.
    2. The force signal goes above the specified minimum force platform value.
3. If "Use Pattern Recognition to Create (L/R)HS and (L/R)TO labels" is checked, and if an ON/OFF event was created in a trial, a HS/TO event will be created in the trial based on the position of the foot segment: (See Event_TPR_Signal section for more information on this)
    1. HS events will be created based on the Axial and AP position of the proximal end of the foot
    2. TO events will be created based on the Axial and AP position of the distal end of the foot

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

Older Versions	`Automatic_Gait_Events`
	`! /SELECT_X=FALSE`
	`! /SELECT_Y=FALSE`
	`! /SELECT_Z=TRUE`
	`! /FRAME_WINDOW=8`
	`! /USE_TPR=TRUE`
	`;`

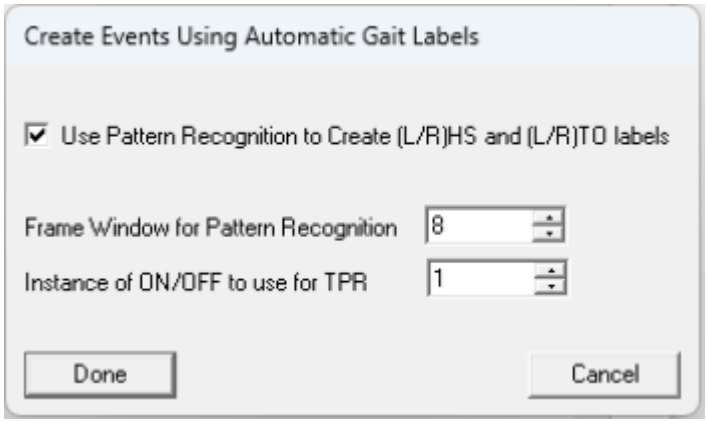
Command Parameters

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

Parameter	Parameter Description
! /FRAME_WINDOW = 8	The frame window used for pattern recognition (default = 8)
! /USE_TPR=TRUE	Enables TPR Algorithm to be used to find the heelstrike (HS) and toeoff (TO).
! /TPR_EVENT_INSTANCE=1	Which instance to use to detect events using TPR.

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 how many instances 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.

Example 1: Simple Scenario

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**, with all the gait events the program automatically detected.

Example 2: Start-to-Finish

In the next example, the **Automatic_Gait_Events** command will be used with other commands to allow a user to go from starting with empty workspace to finding and highlighting events.

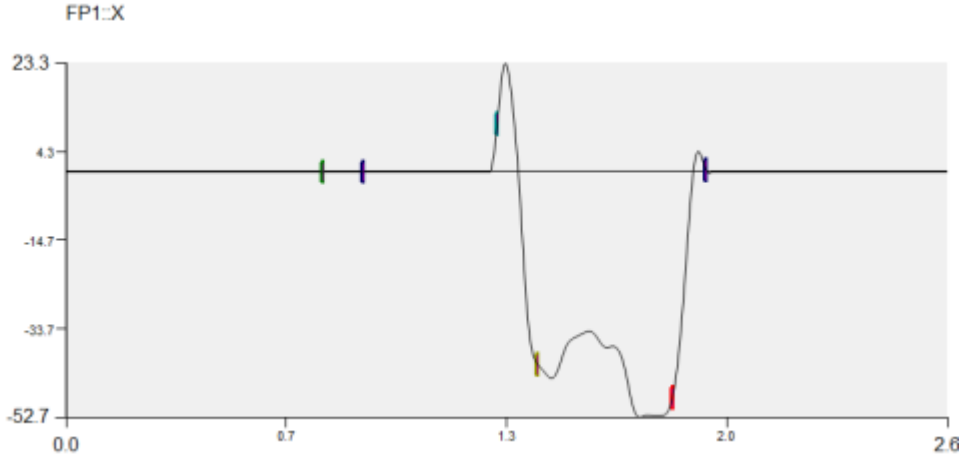
- Open the user-selected workspace file. (File_Open)
- **Automatically detect gait events.**
- Iterate through and highlight all gait events. (For_Each) → (Highlight_Event_Label)
- Switch to Signals and Events tab, and graph X-component of Force Platform signal (events will be shown on this graph). (Switch_Between_Tabs) → (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 run this pipeline, the program is able to display the following graph:



Example 3: Mimicking Automatic_Gait_Events

To show the simplicity of this command, an example pipeline will be shown below to mimic the same functionality using other existing commands, however, it will not necessarily produce the precise TPR created events.

Creating events from kinematic data is accurate to approximately plus/minus 2 frames. In this example, the subject is walking in the +Y direction of the laboratory. The Automatic_Gait_Events command allows Visual3D to determine the direction of walking from the data.

```
For_Each
/Iteration_Parameter_Name=FOOT
/Items=LFT+RFT
;

Conditional_Statement
/ITERATION_PARAMETER_NAME=LEFT
! /EXPRESSION=::FOOT == "LFT"
! /AS_INTEGER=TRUE
;
    Set_Pipeline_Parameter
    /PIPELINE_PARAMETER_NAME=ON_EVENT
    /VALUE=LON
    ;
    Set_Pipeline_Parameter
    /PIPELINE_PARAMETER_NAME=OFF_EVENT
    /VALUE=LOFF
    ;
    Set_Pipeline_Parameter
    /PIPELINE_PARAMETER_NAME=HS_EVENT
    /VALUE=LHS
    ;
    Set_Pipeline_Parameter
    /PIPELINE_PARAMETER_NAME=TO_EVENT
    /VALUE=LTO
    ;
Conditional_Statement_End
/ITERATION_PARAMETER_NAME=LEFT
;

Conditional_Statement
/ITERATION_PARAMETER_NAME=RIGHT
! /EXPRESSION=::FOOT != "LFT"
! /AS_INTEGER=TRUE
;
    ! Right foot events
    Set_Pipeline_Parameter
    /PIPELINE_PARAMETER_NAME=ON_EVENT
    /VALUE=RON
    ;
```

```
    Set_Pipeline_Parameter
    /PIPELINE_PARAMETER_NAME=OFF_EVENT
    /VALUE=ROFF
    ;
    Set_Pipeline_Parameter
    /PIPELINE_PARAMETER_NAME=HS_EVENT
    /VALUE=RHS
    ;
    Set_Pipeline_Parameter
    /PIPELINE_PARAMETER_NAME=TO_EVENT
    /VALUE=RTO
    ;
Conditional_Statement_End
/ITERATION_PARAMETER_NAME=RIGHT
;

Event_Threshold
/RESULT_EVENT_NAME=&::ON_EVENT
/SIGNAL_TYPES=KINETIC_KINEMATIC
/SIGNAL_FOLDER=::FOOT
/SIGNAL_NAMES=FORCE_1
/SIGNAL_COMPONENTS=Z
/THRESHOLD=10
/ON_ASCENT=TRUE
/ON_DESCENT=FALSE
/ENSURE_FRAMES_AFTER=TRUE
;

Event_Threshold
/RESULT_EVENT_NAME=&::OFF_EVENT
/SIGNAL_TYPES=KINETIC_KINEMATIC
/SIGNAL_FOLDER=::FOOT
/SIGNAL_NAMES=FORCE_1
/SIGNAL_COMPONENTS=Z
/FRAME_OFFSET=-1
/THRESHOLD=10
/ON_ASCENT=FALSE
/ON_DESCENT=TRUE
/ENSURE_FRAMES_AFTER=TRUE
;

Event_TPR_Signal
/SIGNAL_TYPES=KINETIC_KINEMATIC
/SIGNAL_FOLDER=::FOOT
/SIGNAL_NAMES=ProxEndPos
/PATTERN_FILE=Rotate Through Active Files
/PATTERN_TYPE=KINETIC_KINEMATIC
/PATTERN_FOLDER=::FOOT
/PATTERN_SIGNAL=ProxEndPos
/PATTERN_EVENT_NAME=&::ON_EVENT
```

```
/SELECT_X=TRUE
/SELECT_Z=TRUE
/TPR_WINDOW=8
/EVENT_NAME=&::HS_EVENT
;

Event_TPR_Signal
/SIGNAL_TYPES=KINETIC_KINEMATIC
/SIGNAL_FOLDER=::FOOT
/SIGNAL_NAMES=ProxEndPos
/PATTERN_FILE=Rotate Through Active Files
/PATTERN_TYPE=KINETIC_KINEMATIC
/PATTERN_FOLDER=::FOOT
/PATTERN_SIGNAL=ProxEndPos
/PATTERN_EVENT_NAME=&::OFF_EVENT
/SELECT_X=TRUE
/SELECT_Z=TRUE
/TPR_WINDOW=8
/EVENT_NAME=&::TO_EVENT
;

End_For_Each
/Iteration_Parameter_Name=FOOT
;
```

Notes

Gait Event Acronyms

Definitions of gait event acronyms used in Visual3D are shown below.

The automatic gait events command, many of the events created in the tutorials, and events used in the example files have a 3 (or 4) character label that have been used consistently throughout history. The labels are divided into two categories:

1. Kinematic Events:

```
**RHS**= Right Heel Strike
**RTO**= Right Toe Off
**LHS**= Left Heel Strike
**LTO**= Left Toe Off
```

2. Kinetic Events:

```
**RON**= Right On
**ROFF**= Right Off
**LON**= Left On
**LOFF**= Left Off
```

All RON events are also RHS events, but RON events are only created when contact is made with a force platform. This provides a means to declare a range of data for reporting only when in contact with the force platform. This is because Joint Moments and Joint Powers, for example, do not have meaningful data when the foot is in contact with ground but no ground reaction forces are measured.

Event_TPR_Signal

A description of the use of Event_TPR_Signal for determining gait events when a force platform signal is present is presented in Stanhope et al.'s paper “A Kinematic-Based Technique for Event Time Determination During Gait.”[1)]

Event_TPR_Signal for the Automatic Gait algorithm uses the Axial and AP trajectory of the proximal end of the foot segment for detecting **Heel Strike**, while using the distal end of the foot segment for detecting **Toe Off**.

- **Heel Strike**: uses the signal KINETIC_KINEMATIC::RFT::ProxEndPos and KINETIC_KINEMATIC::LFT::ProxEndPos
- **Toe Off**: uses the signal KINETIC_KINEMATIC::RFT::DistEndPos and KINETIC_KINEMATIC::LFT::DistEndPos

The HS and TO pattern recognition algorithms are not a generic one size fits all gait recognition algorithm, but rather it looks at the first ON and OFF events and matches the subject's specific pattern at the heel and toe to find other instances of the subject/trial specific pattern within a trial. The algorithm works very well, but if the pattern changes within a trial then the different HS or TO may not be identified.

Working with Multisubject Data

Visual3D handles multisubject data by using the subject prefixes in the C3D file to distinguish which data belongs to which subject. In this case the automatic gait event labels will also have the prefix of the subject to which the events belong.

NOTE: When using events in a command, the commands will iterate across subjects contained in the current workspace. As such, events and sequences listed as command parameters should NOT contain a prefix. As each subject is processed, the event range/sequence specified will automatically use the specific events prefixed for each subject as they are processed.

1)

Stanhope SJ, Kepple TM, McGuire DA, Roman NL. (1990) “A Kinematic-Based Technique for Event Time Determination During Gait.” Medical and Biological Engineering and Computing 28:355-360.