Modules and Messages

The *gesture-follower* is designed as different modules, allowing for rapid prototyping of experiments, and to facilitate the integration with other components. These modules are described below:

Figure 1 Gesture Follower Modules

Module name	Description
`mnm.follower.hmm.pat`	Perform the *learning* and *decoding* phases, and temporarily stocking the data. This abstraction requires: `mnm.follower.makeL2R.pat`, `mnm.follower.L2Rdecoding.pat` `mnm.follower.database.pat`
`mnm.follower.makeL2R.pat`	Create a gesture model, basically a left-to-right Markov Model, during the *learning* mode. The model parameters are stored in a *mat*
`mnm.follower.L2Rdecoding.pat`	Compute the HMM decoding during the play mode.
`mnm.follower.database.pat`	Store all the gestures and decoding parameters in a mat.
`mnm.follower.init.pat`	Initialisation of: `mnm.follower.hmm.pat` `mnm.follower.gui.pat`, `mnm.follower.markers.pat` `mnm.graphs.gui.pat` objects
`mnm.follower.markers.pat`	Enabling the use of *markers* and *envelopes* in the gesture follower. A basic editor for graphically edit the *markers* and *envelopes*
`mnm.follower.graphs4.pat`	Data stream and real-time warping visualization tool
`mnm.follower.gui.pat`	General GUI
`mnm.follower.monitor.pat`	Tool for basic visualization of the decoding process

Description of the Gesture-Follower modules

Messages between modules

Note: All message start with the argument #1 which allows different gesture-followers to run in parallel using different #1 values. This value is the first argument of the following objects:

- `mnm.follower.hmm.pat`
- `mnm.follower.database.pat`
- `mnm.follower.database.pat`
- `mnm.follower.init.pat`
- `mnm.follower.markers.pat`
- `mnm.follower.graphs.pat`
- `mnm.follower.gui.pat`
- `mnm.follower.monitor.pat`

Categories	Message Name	Type or Argument	Description
General	#1_WhatToDo	StartLearn, StopLearn StartDecode, StopDecode	Basic commands to operate the *learning* and *decoding* phases
	#1_Start	0, 1	Start/Stop the gesture for both the learning and decoding mode
	#1_Learn	0, 1	Arm the learning mode (must be activated by Start)
Initialisation	#1_Init	Bang	This bang is the last massage sent by mnm.follower.init.pat. Used to control initialisation process (avoid errors at initialisation)
Database	#1_Section	*Int* Clear	*Select the Section number, clear the memory of all the sections*
	#1_Phrase	*Int* Clear	*Select the Section number, Clear the memory of all the phrases*
Learning input parameters Learning output parameters	#1_FollowerDatabase	Mat methods (#save, #load, #open, etc) Update	Access to the mat containing all the gesture data Update the current state of the patches with the content of the mat. Required after #load
	#1_Gestures	*mat*	mat containing all the data
	#1_HMMs	*mat*	mat containing the *HMMs* of a given section
	#1_Resample	float	Resampling value used at the learning phase. This value set the maximum speed during the play mode
	#1_newHMM	dict	HMM corresponding to a given phrase of a given section
Decoding input parameters	#1_StdDev	float	Expected average intensity difference between learning and play modes
	#1_Cyclic	0, 1 = cyclic	1= the last state of *HMMs* is lopped to the first one.

Decoding output parameters from mnm.follower.hmm.mat	#1_Weight	*fmat*	Weight the importance of each channel
	#1_Contrast	float	Likelihood values can be modified using an exponential scaling. This allow for the increase or decrease the difference between the values
	#1_LHthreshold	float]-∞... 0]	Threshold for restarting automatically the decoding
	#1_Index	*fmat* [0...1]	Index of most likely state , respectively for all *HMMs*: *fmat*
	#1_IndexMax	*fmat* (*Int*s)	Number of state,
Decoding output parameters from mnm.follower.markers.pat	#1_LH	*fmat*]-∞... 0]	Likelihoods, respectively for all *HMMs*: *fmat*
	#1_LHnorm	*fmat* [0... 1]	Normalized likelihoods, respectively for all *HMMs*: *fmat*
	#1_LHmax	*fmat*]-∞... 0]	Maximum of likelihood values
	#1_Likeliest	*Int*	Likeliest phrase
	#1_State	*fmat*	Most likely state, respectively for all *HMMs*: mat
	#1_Markers	*fmat* (*Int*s)	Markers number
	#1_Enveloppes	*fmat*	Envelope values

1.1.1 Interfaces of the developed modules

mnm.follower.hmm.pat		
Description	Abstraction performing the learning and decoding of the HMM	
Instantiation Arguments	*Int: index starting all messages*	
Embedded Externals	`mnm.follower.makeL2R.pat` `mnm.follower.L2Rdecoding.pat`	
Attributes		
Methods		
Inlets	inlet 1	*Data stream: fmat (row vector size 1 N)*
Messages received	#1_WhatToDo	
	#1_Resample	
	#1_HMMs	
	#1_StdDev	
	#1_Cyclic	
	#1_Weight	
	#1_Contrast	
	#1_LHthreshold	
Messages sent	#1_newHMM	
	#1_Index	
	#1_IndexMax	
	#1_LH	
	#1_LHnorm	
	#1_LHmax	
	#1_Likeliest	
	#1_State	

mnm.follower. makeL2R.pat		
Description	Abstraction creating the HMM model	
Instantiation Arguments		
Embedded Externals		
Attributes		
Methods		
Inlets	inlet 1	*fmat (row vector size 1 N)*
	inlet 2	Resample value (*float).* *This value set the maximum speed in the play mode*
Outlets	outlet 1	*HMM model (dict)*
Messages received		
Messages sent		

mnm.follower.L2Rdecoding.pat	
Description	Abstraction performing the decoding of the HMM model
Instantiation Arguments	

Embedded Externals		
Attributes		
Methods		
Inlets	inlet 1	*Data stream: fmat (row vector size 1 N)*
	inlet 2	*Start:Bang*
	inlet 3	*Average Standard Deviation StdDev: float. Expected average difference between phrases in learning and play mode.*
	inlet 4	*Cyclic 0/1. 1= the last state of HMMs is lopped to the first one.*
	inlet 5	*Weight: fmat (row vector size 1 N). Weight the importance of each channel*
	inlet 6	*HMMs: mat containing all HHMs models*
Outlets	outlet 1	*Most likely state, respectively for all HMMs: mat*
	outlet 2	*Likelihoods, respectively for all HMMs: fmat*
	outlet 3	*Index of most likely state , respectively for all HMMs: fmat*
	outlet 4	*Number of state,*
Messages received		
Messages sent		

mnm.follower.database.pat		
Description	Abstraction storing all the gesture database	
Instantiation Arguments	*Int: index starting all messages*	
Embedded Externals		
Attributes		
Methods		
Inlets		
Outlets		
Messages received	#1_newHMM	
	#1_Section	
	#1_Phrase	
	#1_Weight	
	#1_FollowerDatabase	
Messages sent	#1_HHMs	
	#1_Section	
	#1_Phrase	
	#1_Weight	

mnm.follower.init.pat		
Description	Abstraction performing the initialisation of the patch parameters	
Instantiation Arguments	*Int: index starting all messages*	
Embedded Externals		
Attributes		
Methods		
Inlets	inlet 1	*Gesture Follower Parameters (Dict)*
	inlet 2	*Reset (bang)*

	inlet 3	*Access to the dict (#open, #save, #load, #print, etc)*
Messages received		
Messages sent	#1_HMMs	
	#1_Phrase	
	#1_Section	
	#1_Contrast	
	#1_LHthreshold	
	#1_Cyclic	
	#1_StdDev	
	#1_Resample	
	#1_Init	

mnm.follower.markers.pat		
Description	Abstraction for the editing and visualization of *markers* and *envelopes*	
Instantiation Arguments	*Int*: index starting all messages, *Int: ch1, Int:* ch2*, Int:* ch3	
Embedded Externals		
Attributes		
Methods		
Inlets	inlet 1	*Bang,* clock needed for the realtime acquisition of markers and envelopes
	inlet 2	*Float, envelopes y-input*
Outlets	outlet 1	*Int: marker number*
	outlet 2	*Float: envelope*
	outlet 3	*Float x-axis of display*
Messages received	#1_WhatToDo	
	#1_Phrase	
	#1_HMMs	
	#1_Likeliest	
	#1_Index	
	#1_IndexMax	
	#1_init	
Messages sent	#1_Markers	
	#1_Envelope	

mnm.follower.hmm.pat		
Description	Abstraction performing the learning and decoding of the HMM	
Instantiation Arguments	*Int: index starting all messages*	
Embedded Externals	`mnm.follower.makeL2R.pat` `mnm.follower.L2Rdecoding.pat`	
Attributes		
Methods		
Inlets	inlet 1	*fmat (row vector size 1 N)*
Messages received	#1_WhatToDo	
	#1_Resample	
	#1_HMMs	
	#1_StdDev	
	#1_Cyclic	

	#1_Weight	
	#1_Contrast	
	#1_LHthreshold	
Messages sent	#1_newHMM	
	#1_Index	
	#1_IndexMax	
	#1_LH	
	#1_LHnorm	
	#1_LHmax	
	#1_Likeliest	
	#1_State	