Score Following: Artificially Intelligent Musical Accompaniment

Anna Jordanous

Music Informatics
University of Sussex

This talk will cover:

- Accompaniment as an Artificial Intelligence task
- Background
  - What is score following?
  - Previous work in score following
  - Using Hidden Markov Models
- Producing an artificial accompanist
  - Design and Implementation details
  - Evaluating the performance of the artificial accompanist
- Inspiration for further work

What is score following?

Score
The written music that
a musician reads
when they play music

Score Following
Following a musician's
performance of a piece
by tracking their playing
through the score

When do we use Score Following?

Imagine you are at a concert. A flute player is performing a solo piece, with a piano player providing accompaniment.

To make sure their accompaniment matches the flute player, the pianist listens to what the flute player is playing, following their playing through the score of the piece.

The flute player may occasionally not perform the piece exactly as it is written in the score. In these cases the pianist must adjust their accompaniment.

Why might the flute player not perform the piece exactly as written?

The flute player may make mistakes…

- Play the wrong note
- Add extra notes
- Skip notes out
- Lose their place in the music
- Speed up or slow down accidentally

… or they may make changes deliberately, to add their own interpretation to the music

- Add embellishments such as trills, to decorate the notes
- Speed up or slow down deliberately, for musical effect

NB. These are not exhaustive lists!

Score Following is vital for artificially intelligent accompaniment

Accompaniment is not just about playing the accompaniment music accurately.

The accompanist must play the right accompaniment at the right time, following the soloist's performance.

Score following allows us:

> *to recognise the computer's potential ... as an intelligent and musically informed collaborator in live performance*
>
> *Vercoe (1984)*

Research into score following: a timeline

1984 — Using dynamic programming* to compare the soloist's playing with what they were expected to play (Vercoe; Dannenberg)

*(*Dynamic programming = solving problems recursively by continuously breaking the problem down into smaller subproblems)*

1985 — Vercoe adds learning to his system (to learn the performer’s likely paths through the score)

1989 — Scheduling (Dannenberg) Real time (clock) vs Synthetic time (performer's perceived time)

1998 — Probability density functions (Grubb & Dannenberg) Estimating current location in score using probability

1999 — First use of Hidden Markov Models to statistically model the music (Raphael; Cano, Loscos and Bonada)

2001 — IRCAM present their initial work on score following, trying different methods before settling on Hidden Markov Models.

Raphael adds Bayesian probability to his score followers (probability of reaching a point in the score given previous play)

2005 — A comparison of Hidden Markov Models against other methods finds that Hidden Markov Models perform better in jazz accompaniment (Pardo & Birmingham

present

Hidden Markov Models are increasingly being used in score following

Many research efforts focus on using Hidden Markov Models

- Raphael 1999 onwards
- Cano, Loscos and Bonada 1999
- IRCAM 2001 onwards
- Pardo and Birmingham 2005

What are Hidden Markov Models ?

How are they useful for artificial accompaniment ?

A Hidden Markov Model is a way of modelling real-world scenarios

A real-life system can be modelled with an HMM if:

- The system can be thought of as being in one of a (finite) number of *states*, at any given time.
- We can determine what state the system is currently in by looking at a sequence of recent outputs from the system (*observations*).

e.g. Flipping coins a number of times - where the coin being flipped is either a biased or normal coin (but we don't know which one it is and want to try and find out).

STATES: 'biased coin flipped', 'normal coin flipped'
OBSERVATIONS: 'H', 'T'.

By looking at the sequence of results e.g. H, T, T, T, H, H

Can we work out which coin is being used?
... even if we allow the coin to be changed between flips?

A Hidden Markov Model is a way of modelling real-world scenarios

A Hidden Markov Model (HMM) uses probabilities relating to these states and observations:

- Probability of the system initially starting in a particular state
- Probability of various observations happening while the system is in a particular state
- Probability of the system moving from being in one state to being in another

A Hidden Markov Model is a way of modelling real-world scenarios

P(initially starting with the biased coin) = 0.5
P(initially starting with the normal coin) = 0.5

P(head | normal coin is being used) = 0.5
P(tail | normal coin is being used) = 0.5
P(head | biased coin is being used) = 0.1
P(tail | biased coin is being used) = 0.9

P(change from using the normal coin to the biased coin) = 0.3
P(change from using the biased coin to the normal coin) = 0.05

Score Following can be done using a Hidden Markov Model

A music score can be divided into a sequence of *events, such as*

- Notes
- Beats
- Cadences
- Phrases
- Chords
- etc

A score follower can determine what state the performer is most likely to be in at that time, by:

- Looking at the *sequence of events* actually being performed by the soloist
- Matching that sequence of events to the Hidden Markov Model of the score

Modelling a score using a Hidden Markov Model: an example

STATES: {Beat1, Beat2, Beat3, Beat4, Beat5, Beat6, Beat7, Beat8}

OBSERVATIONS: The note being played by the soloist at that point in time

How would probabilities be allocated?

Score Following can be done using a Hidden Markov Model

Problem:

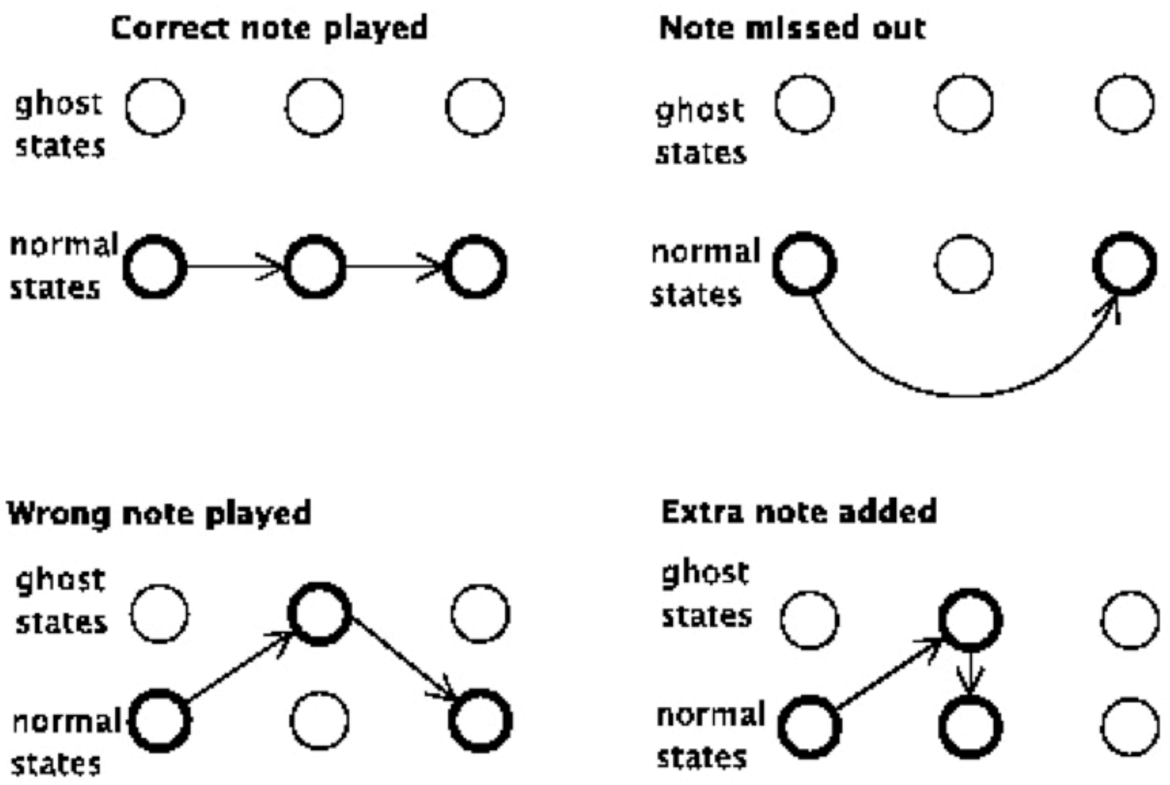
What about when the performer deviates from the score?
How can that be represented in a HMM? We can't possibly come up with a state that represents every possible deviation from the score

Orio et al. came up with a solution for this problem in their 2003 paper.

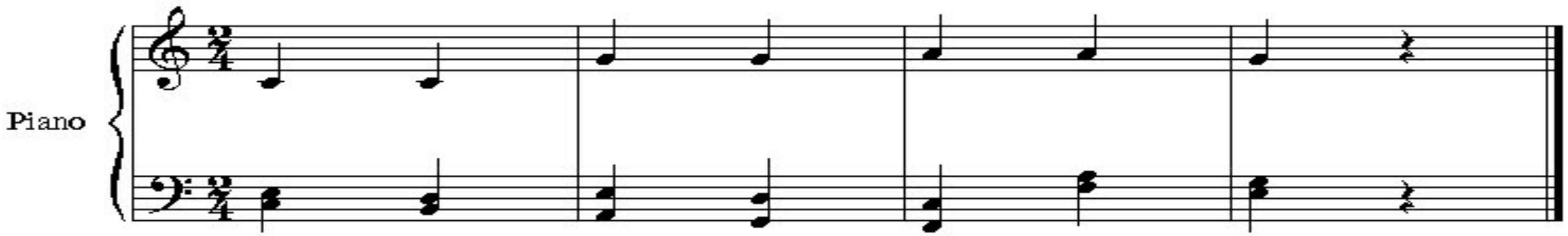
For each event in a score, use <u>two</u> HMM states:

- 'Normal' state – for when the performer plays the expected note
- 'Ghost' state – for when the performer plays an unexpected note

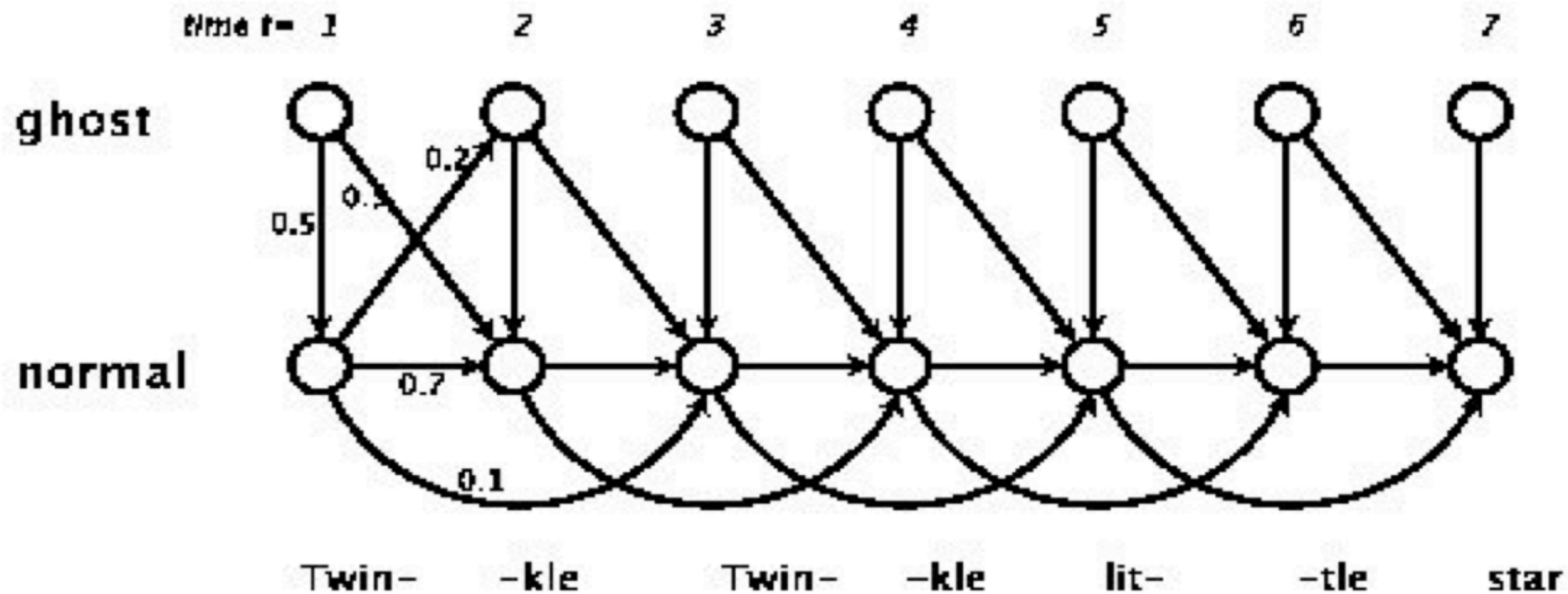
Normal and Ghost states can be used to represent deviations from the score



Back to the Twinkle Twinkle example…



Back to the Twinkle Twinkle example…



Making an artificially intelligent score follower using HMMs

The hypothesis being tested is:

Using an HMM to model a musical score is an efficient and practical way to implement score following.
In particular, it lends well to providing real-time accompaniment to a human performer.

To model the scores, I chose to use the HMM states to represent individual beats of the piece:

A piece of music can be thought of as a sequence of beats, with certain notes expected at each beat (i.e. the note written in the score)

Construction of an artificially intelligent score follower using HMMs

Written in Max/MSP

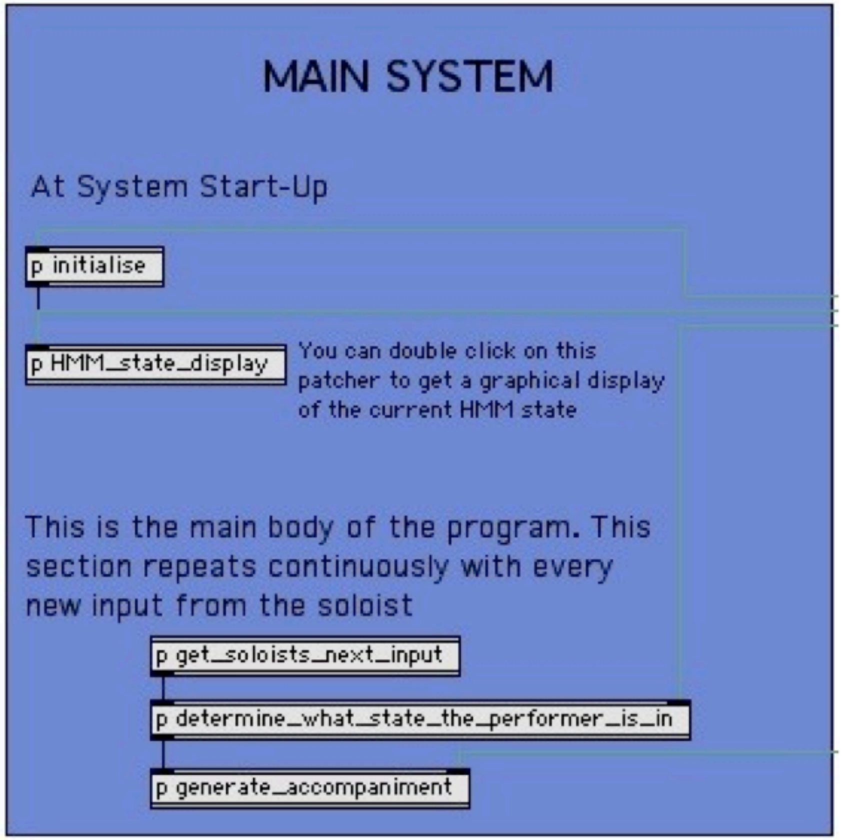
Max/MSP is a real-time, interactive music programming environment

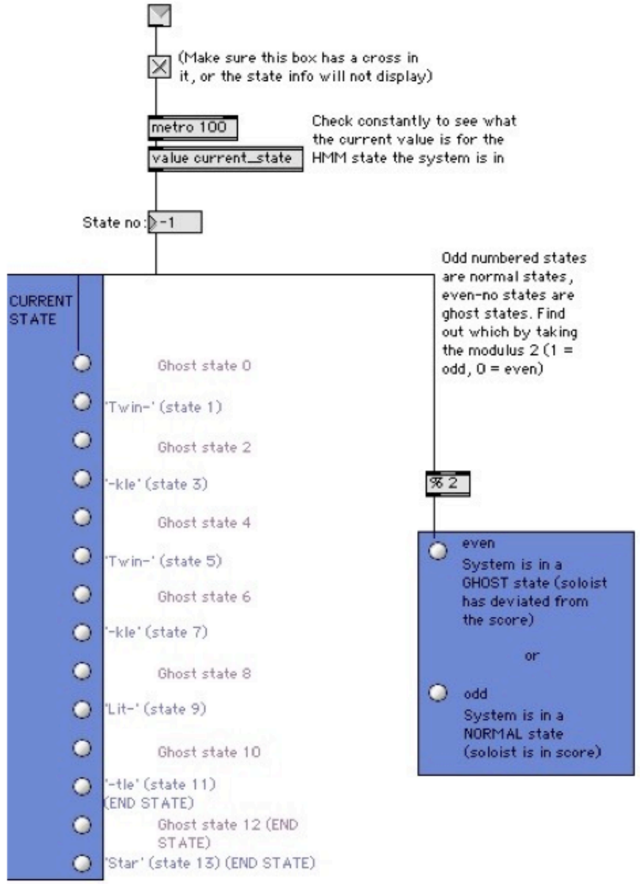
The score follower runs on a Mac or PC computer that is attached to a MIDI keyboard

The MIDI keyboard communicates with the computer using the musical format MIDI

MIDI messages = coded information about the notes to be played:
pitch, volume, onset time, output channel etc

The HMM Score Follower System





The program works by continuously repeating three basic steps

get_soloists_next_input:

Extracts the new information from MIDI keyboard.

determine_what_state_the_performer_is_in:

Keeps a list of observations seen so far. Performs the Viterbi decoding algorithm to use the most recent observations to work out which HMM state the soloist is in (i.e. where they are in the score).

generate_accompaniment:

Looks up the HMM state in an attribute-value pair, to find which notes to play as accompaniment

E.g: <1, [48, 52, 55]> - 'In state 1, play MIDI notes 48, 52, 55' (C major chord)

While performing accompaniment, the system also tracks the performer's tempo

Beat tracking is the process of working out the tempo of a piece of music by analysing what is being heard.

I use very simple methods in comparison to the latest state of the art beat-tracking research! (Gouyon and Dixon, 2006)

- Measure time in between notes played by the soloist and use this to estimate the soloist's tempo
- If the soloist is currently in a *ghost* state (they have deviated from the score) then *ignore* the notes they are currently playing, when estimating the tempo

The score follower has been evaluated quantitatively and qualitatively

Evaluated using both objective measurements and subjective judgements of performance, using three melodies of increasing complexity.

Quantitative testing:

- Using criteria devised to test score following systems at **M**usic **I**nformation and **R**etrieval **EX**change conference in 2006
- Measure % of correctly played accompaniment and timing issues

Qualitative testing:

- Four testers of varying musical ability were asked to test the system
- Asked to freely experiment with the system as they saw fit

Comparing my score follower to systems tested at MIREX 2006

Total precision: % of correctly detected score notes overall (across all pieces)
***Piecewise precision**: % of correctly detected notes, averaged across each piece*

Cont and Schwarz (MIREX 2006) - Using Hidden Markov Models

Total precision: 82.90 %
Piecewise precision: 90.06 %

Miller Puckette (MIREX 2006) - Using Dynamic Programming

Total precision: 29.75 %
Piecewise precision: 69.74 %

This score follower - Using Hidden Markov Models

Total precision: 60.89 %
Piecewise precision: 54.04 %

Results of testing

- Very accurate performance for the simpler melodies reported by both objective measurements and tester feedback
  - Accompaniment was played accurately even when the performer deviated extensively from score
  - No latency issues *(latency = delays in playing)*
- But: pretty poor performance in the more complex melodies
  - The score follower did not locate the performer's position in the score as well, performing with much lower precision
  - Latency became a big issue with longer pieces – accompaniment was played with delays of 200-900 ms

Some interesting aspects were noted during evaluation

- The most complex piece had a large number of repeated phrases and similar sections, confusing the score follower somewhat.
  - Increasing the number of observations used by the score follower increased the level of accuracy
- Delays in accompaniment for the longer pieces was mostly due to high computation load for each note the performer played. Latency was reduced considerably by:
  - Calculating HMM-based probabilities in advance for each state, rather than making all calculations while the program was running
  - Considering only a sub-set of all the possible locations in the score – those local to the current position
- The score follower was too reliant on following the performer and did not have 'confidence' in playing what it perceived to be right
  - This is not addressed in score following research to date but Dannenberg has commented on the role of the computer performer in ensemble situations. Also there is interesting related research in *entrainement (musical synchonisation)*.

Summary: Artificially Intelligent Accompaniment using HMMs

- Accompaniment is a task which requires musical intelligence
- Automated musical accompaniment, or score following, can be implemented using Hidden Markov Models (HMMs) to track what state a performer is in
- Some useful results were obtained for the accompaniment of simple melodies, however accuracy and latency issues affected performance of longer and more complex/repetitive melodies
- Problems in initial results could be addressed to some extent, however there is still a high computational cost in using HMMs for score following that needs to be considered
- This research casts interesting questions on the seemingly passive role of the accompanist in a perfomance scenario
- Lets hear the system in action (warts and all...)