# Gaussian Process Inference Modelling of Dynamic Robot Control for Expressive Piano Playing

## The Motivation

In the development of robots that perform music, one of the greatest challenges faced is in engaging the audience with expressive and emotive performances. We can achieve expressive music performance through variation in the timing, volume and articulation of each musical note. To do this, robots need to be able to listen to themselves playing, learn from mistakes and adjust their movements and techniques to improve accordingly.

## The Goal

This project aims to develop a robotic system that changes its actions and playing techniques based on audio feedback. To evaluate how effective our method is, we can then test its ability to accurately reproduce the sounds in reference audio samples.

## Experimental Setup

In this project, monophonic music is used, which requires only single-finger keystrokes. Our simple setup in Figure 1 consists of a UR5 robot arm, a flexible "finger" 3D-printed using soft material (TPE), and a digital piano that gives MIDI format audio data.

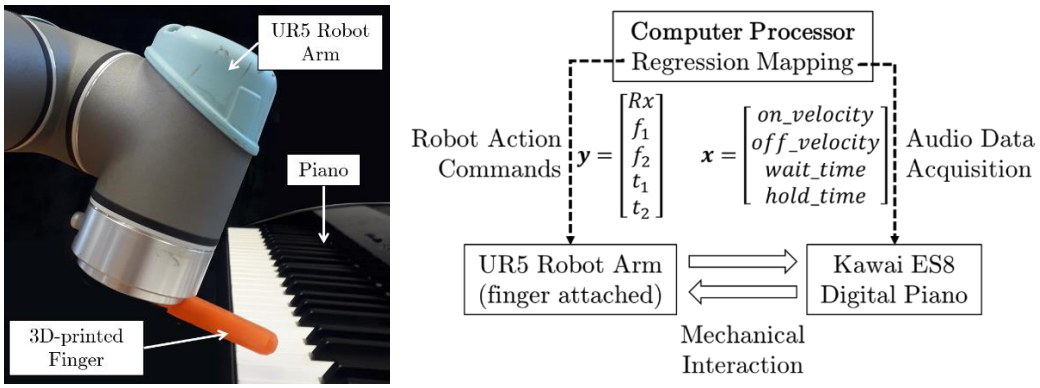


*Figure 1 Experimental setup and a schematic of the robot connection to a processing unit and the piano.*

## Data

We deal with two types of data: audio data and control parameters. The audio data, $\boldsymbol{x} = [on\_velocity, off\_velocity, wait\_time, hold\_time]$ are digital values obtained from the piano for every key-stroke, while the control parameters, $\boldsymbol{y} = [Rx, f_1, f_2, t_1, t_2]$ set the robot's actions for each key-stroke.

*Audio Data*

MIDI format audio messages are generated when a piano key is pressed or released to give the four variables in Figure 2: $on\_velocity$ is the note's amplitude when its key is pressed and $off\_velocity$ the abruptness of the release of the key, which correspond to the downward and upward velocity registered by the piano key's sensors. $hold\_time, wait\_time$ are how long the current key is held down for and the time until the next key is pressed respectively.

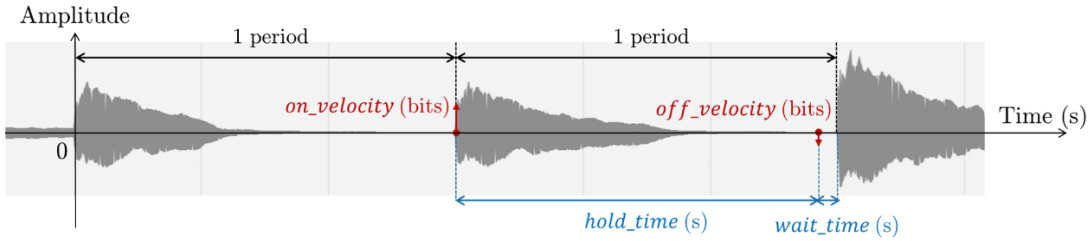


*Figure 2 How the MIDI audio variables relate to a monophonic piano audio signal in time domain.*

*Control Parameters*

The robot performs each parameterised key-stroke with its attached finger in 4 steps (a) – (d) (Figure 3). In (a), it moves from one piano key position to another and rotates $A_{Rx} = Rx$ angles about the x-axis at the tool centre point (TCP), within a specified $t_1$ seconds. In (b), the finger presses downwards onto the key following a sinusoidal path of frequency $f_1$ Hz (Figure 4). In (c), the finger holds at the pressed position for a specified $t_2$ seconds. In (d) it releases the key upwards following a sinusoidal path of frequency $f_2$ Hz. The variables $Rx, f_1, f_2, t_1, t_2$ therefore change the way the finger interacts with the piano key and the corresponding audio data generated.

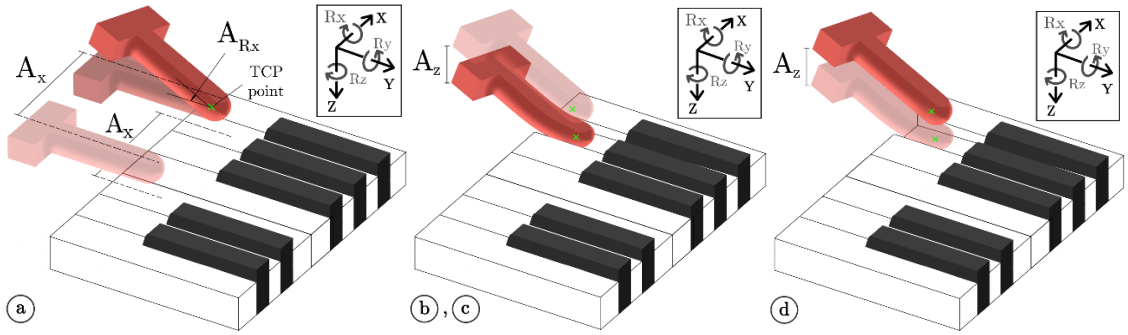


*Figure 3 Illustration of the finger's movement during a key-stroke (a) - (d).*

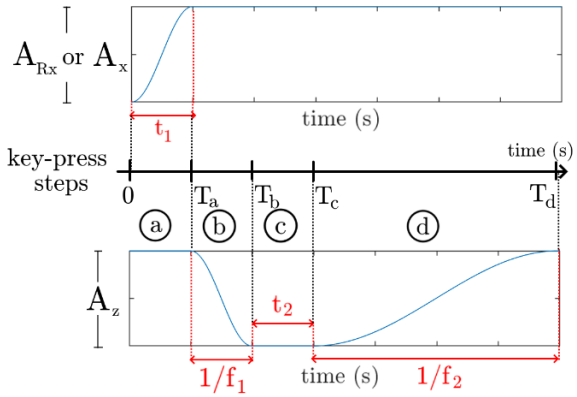


*Figure 4 Sinusoidal robot control in the x and z-axis respectively.*

**Data Relationships**

In the experiments, we collected 3125 datapoints using different control parameter values. Figure 5 plots examples of the relationship between one control variable and one audio variable. As seen from the curved trendlines, our data is non-linear (cannot be modelled by a

straight line "$y = mx + c$"). As the shape of the trendlines are different when the second dependent variable, $Rx$ changes, our data also contains cross-correlations.

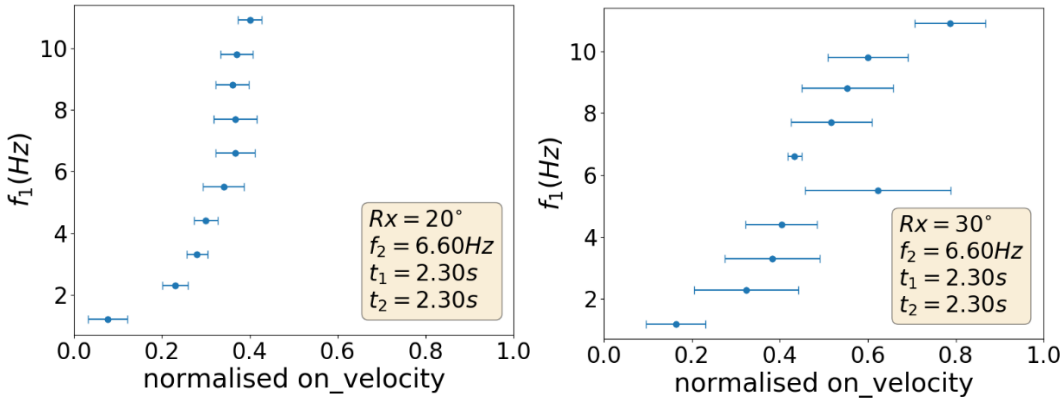


*Figure 5 Example Plots of $f_1$ vs on_velocity at different Rx angles.*

## Modelling the Data

Multi-variate regression is a technique to estimate the relationships between multiple dependent and independent variables in a dataset. It is important to fit a suitable regression model to our dataset so that it can be used to predict what control parameters are suitable to reproduce audio samples.

We use Gaussian processes (GPs) as it is suitable for our non-linear and cross-correlated dataset. Unlike models like polynomial regression (e.g. $y = ax\text{^}2 + bx + c$), a Gaussian process models the relationship as a probability normal distribution at each point across the data range, with a mean prediction function and a variance function that indicates how uncertain the model is with its mean prediction at each point. Figure 6 shows how the GP fits the data trend gradually as more datapoints are added, and how the model variance "collapses" as it becomes more certain of the relationship along the range of values.

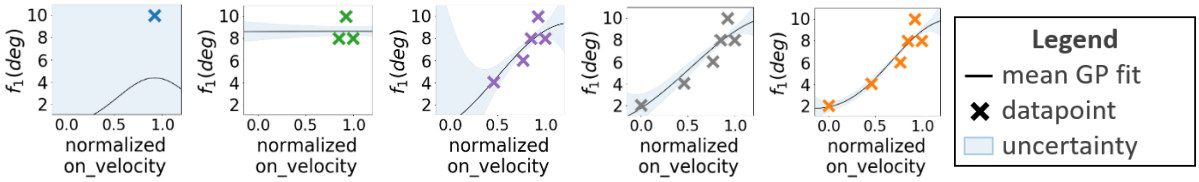


*Figure 6 GP regression fit on 1, 3, 5, 8 and 12 datapoints.*

## Prediction Error on Audio Samples

After training the regression models on 3125 datapoints, we apply them to make predictions on audio samples for 10 playing-styles that were generated in a digital audio synthesizer (digital reference) and by a pianist (human reference). For each style, the audio generated from the prediction, $\boldsymbol{x}_{style}^{inf}$, is compared to the reference sample, $\boldsymbol{x}_{style}^{ref}$, for an error value using $\text{error}_s = \sqrt{\left(\boldsymbol{x}_{style}^{inf} - \boldsymbol{x}_{style}^{ref}\right)^T \left(\boldsymbol{x}_{style}^{inf} - \boldsymbol{x}_{style}^{ref}\right)}$. The normalised errors in both cases are plotted in Figure 7. Here we can make three general observations. Firstly, the errors vary greatly across styles, indicating that the robot can approximate certain playing styles better than others. Secondly, in the left plot we see that the robot performs comparably to the

human in terms of the errors being in the same range of values. In fact, it performs better at styles such as *normal, tenuto, staccatissimo, staccato* and *pppp*. This is largely due to the precision of the robot's control at low speeds compared to the pianist's. However, when we infer on the human reference in the right plot, we see that the robot performs poorly on *staccato*, a style that requires quick key releases and *ff*, a style that requires loud playing. This is because the pianist exaggerates the short and loud notes in the audio samples, making it hard for the robot to approximate as they require extremely quick control in time and velocity.

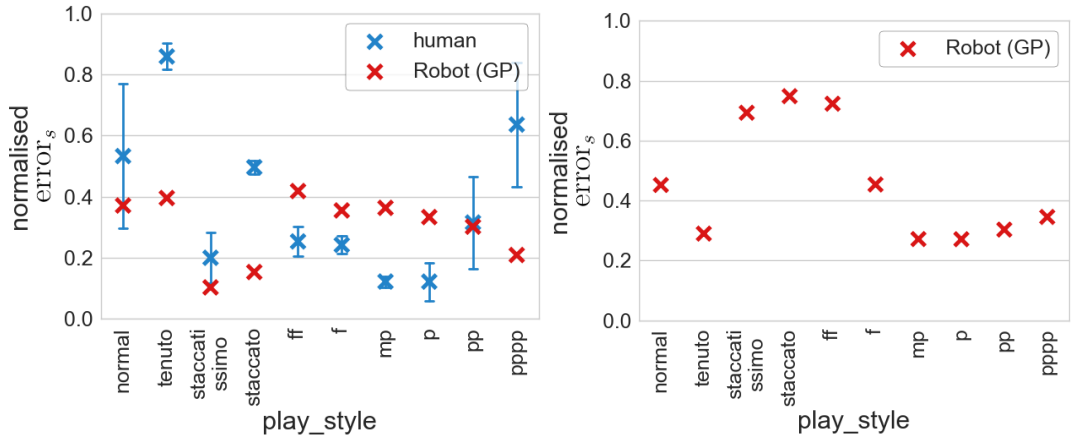


*Figure 7 Audio prediction error for each playing style of a) the human pianist and the trained robot when inferring from the digital reference (left) and b) of the trained robot when inferring from the human reference (right).*

**Conclusion**

We successfully applied Gaussian process regression to model the robot control based on audio feedback and analysed its performance in reproducing audio samples. Future work can include performing more efficient informed exploration of control parameters based on the model's predictive uncertainties and modifying the setup to tackle the physical limits and extend to polyphonic music.

**Additional Resources**

Python code for the project:

https://bitbucket.org/lucascimeca/robo_piano_learning/src/master/

Audio samples: https://clyp.it/user/oufhutxs

Video demonstrations:
https://www.dropbox.com/sh/ub071l78ynird6y/AACnvKlNKpFqWOv4aJkcZ6I-a?dl=0