Hierarchical Multiple Output Gaussian Processes for Human Motion Data

Student: Juan David Gil López

Supervisor: Mauricio Alexander Álvarez López

Universidad Tecnológica de Pereira
Engineering Faculty
Master in Computer Science and Systems Engineering
Research Group in Automática
SIRIUS Research Group
Pereira, Risaralda
2017

Content

1 Introduction **4**

2 Objectives **6**

2.1 General Objective 6

2.2 Specific Objectives 6

3 Background **7**

3.1 Hierarchical Modelling for Gaussian Processes 7

3.1.1 Single output Gaussian Process Regression 7

3.1.2 The hierarchical model 8

3.2 Linear Model of Corregionalization(LMC) 10

3.2.1 Corregionalization Matrix 11

3.3 Computer-based human animation with motion capture 11

3.4 Human Motion Synthesis and Interpolation 12

4 Materials and Methods **15**

4.1 Multiple Output Hierarchical Gaussian Process (MOHGP) 15

4.2 Parameter Estimation Procedure: MOHGP 19

4.3 Synthesis Performance Measures 19

5 Experimental Results and Discussion **22**

5.1 Interpolation 22

5.1.1 Artificial Data 22

5.1.2 MOCAP Data 28

5.2 Synthesis 29

5.2.1 Artificial Data 30

5.2.2 MOCAP Data 31

5.2.3 Parent and children process 35

5.2.4 Coregionalization Matrices over MOCAP data 37

6 Experimental Results Summary 41

6.0.1 Results over interpolation 41

6.0.2 Results over Synthesis 45

7 Conclusions and future work 49

A Appendix 51

A.1 Single Output Hierarchical GP Likelihood Demonstrations 51

A.2 Multiple Output Hierarchical GP Likelihood Demonstrations 52

A.3 Proposed Model: Gradients for the likelihood function (Two layer hierarchy) 53

A.4 Discussion on synthesis evaluation measures 56

A.4.1 Discussion on synthesis measures over artificial data 56

A.4.2 Discussion on synthesis measures over MOCAP data 56

List of Figures

1 Single Output Hierarchical model illustration. 9

2 Performers sensor 12

3 Diagram of proposed model 18

4 Covariance/sample of the parent process 23

5 Covariance/sample from the children process 24

6 Underlying trend prediction 32

7 Synthesis, walking ICM 33

8 Synthesis, walking normal 33

9 Synthesis, walking exaggerated 34

10 Synthesis, soccer shooting 35

12 Estimated parent covariance comparisson 36

13 Estimated parent covariance comparisson 2 36

14 Coregionalization matrices for Walking 38

15 Coregionalization matrices for Walking: left leg 39

16 Coregionalization matrices for Walking: Head and Leg(left femur) 40

17 Results of interpolation: Experiment one on artificial data. 41

18 Results of interpolation: Experiment two on artificial data. 42

19 Results of interpolation: Experiment three on artificial data. 43

20 Results of interpolation: Walking. 44

21 Results of synthesis: Experiment one with artificial data. 45

22 Results of Synthesis: Walking. 46

23 Results of Synthesis: Walking. 47

24 Results of Synthesis: Walking. 48

25 Diagram of an alternative representation for likelihood covariance 54

List of Tables

1 MSE of interpolation by the models under different noise levels 24

2 SMSE of interpolation by the models under different noise levels 25

3 MSLL of interpolation by the models under different noise level 25

4 MSE of interpolation of both models, MOA tuned 25

5 SMSE of interpolation of both models, MOA tuned 26

6 MSLL of interpolation of both models, MOA tuned 26

7 MSE of interpolation of both models, MOA tuned, fixed noise 27

8 SMSE of interpolation of both models, MOA tuned, fixed noise 27

9 MSLL of interpolation of both models, MOA tuned, fixed noise 27

10 Walking: Accuracy performance of interpolation over 2(1) MOHGP (Matern 32 + White Noise) 29

11 Walking: Accuracy performance of interpolation over 2(1) MOA (Walking): (Matern 32 + White Noise) 29

12 MCC of synthesis by the models under different noise levels 30

13 MLC of synthesis by the models under different noise levels 31

14 Walking: MCC and MLC comparison of synthesis. Two versions of the proposed model 32

15 Walking: MCC and MLC comparison of synthesis of both models 32

16 Walking Exaggerated: MCC and MLC comparison of synthesis for both models. In bold letter are highlighted the best results for both models and the average across tests. In red letter are highlighted the worst results of MOHGP against MOA. 34

17 Soccer Shooting: MCC and MLC comparison of synthesis for both models. In bold letter are highlighted the best results for both models and the average across tests. In this case, the best are different from the training subjects. In red letter are highlighted the worst results of MOHGP against MOA. 34

18 Walking: MCC Mock test. 56

19 Walking: MLC Mock test. 57

20 Walking: MCC Mock test. 57

21 Walking: MLC Mock test. 57

Acknowledgements

This work has been supported by the project Human-motion Synthesis through Physically-inspired Machine Learning Models *funded by Universidad Tecnológica de Pereira with code 6-15-3. I would like to thank initially the help and inspiration given by my supervisor, Mauricio, his guidance and classes made me more interested about Machine Learning models and applications. Thanks to him I know now the importance and relevance and potential of this area of research. Also I want to express my gratitude to my familiy, that has been there for me in the most complicated parts of this process, if I would have started my masters back in 2014 without their support, I think it would have been very difficult to overcome some of the impasses along the way. Specially I want to offer special gratitude to Mr. Bill Warner, whose advices and finnancial support in this middle phase of my education have been vital to the pursuit of my carreer objectives. Besides I would like to sincerely recognize the help of the SIRIUS research group, which I belong to, for the access to computer and workspace resources that made less dificult the development and testing of the methodology proposed in this work. Finally thanks to my close friends, without their technical and non-technical tips some of the challenges faced in this thesis would be really difficult to overcome.*

Abstract

Statistical applications usually involve several parameters and observations that sometimes are connected in some way depending on the structure of the problem. Often, these parameters are used as variables that encode some information related with the observations. Besides, these parameters are not observed and not directly measurable, but rather are inferred given the particular correlations between the observed values. Thus, it is natural to model the phenomena with some kind of hierarchical structure in which the observable measures are conditioned to some parameters, and the parameters also conditioned to hyperparameters and so on. These types of models are relevant in the sense that the data prove to be well fitted using that dependence to model the problem. In the case of regression, non-parametric models such as Gaussian Processes (GP) have also been proposed in a hierarchical structure, which depends on the problem to be modelled or the phenomena to be studied.

Different hierarchical approaches have been proposed in the literature. Recently a novel hierarchical modelling for GPs was introduced in which the model assumes there are several observed signals that are related by an underlying trend common to these observations which can be predicted given the data. Thus, the observed signals can be seen as corrupted versions of that underlying trend. Nevertheless, this kind of modelling has only been developed in a single output framework, then it is suitable to explore an extension to multiple output data, given the fact this model has proven to be a simple and powerful tool for analyzing correlations and common trends between several observations of a phenomenon. Thus, in this work is presented an extension of a GP hierarchical model using existing covariances in a multiple output framework to interpolate and synthesize human-like motion data. The model was tested with both artificial and real human motion data. The results show that the model successfully interpolates and synthesizes human motion in comparison to a simple multiple output GP used here as a reference model.

Resumen

Diferentes aplicaciones estadísticas implican el uso de diferentes parámetros y observaciones que en muchos casos están relacionadas de alguna manera dependiendo de la estructura del problema. Usualmente, estos parametros son usados como variables que codifican cierta información relacionada con las observaciones, además, ya estos parametros no son observables ni tampoco pueden ser medidos directamente, son inferidos de los datos observados gracias a las correlaciones dadas entre los mismos. De esa manera, se vuelve natural el modelar el fenómeno por medio de una estructura jerárquica en donde las variables observadas esten condicionadas a los parámetros, y a su vez estos parámetros condicionados a hiperparámetros, etc. Este tipo de modelos son relevantes en el sentido de que sirven cómo buenas aproximaciones al comportamiento de los datos. En el caso de regresión, modelos no paramétricos cómo los procesos Gausianos han sido propuestos también con algún tipo de estructura jerárquica, la cuál depende del problema a ser estudiado.

Diferentes modelos jerárquicos han sido propuestos. Recientemente un novedoso método jerárquico para procesos Gausianos fue propuesto, en dicho modelo, se asumen que existen diferentes señales observadas que están relacionadas por una tendencia común a todas estas observaciones, la cuál puede ser predecida. Así, las señales observadas pueden ser vistas como versiones corruptas de esa tendencia común. Sin embargo, este tipo de modelos solo ha sido desarrollado para modelos de una sola salida, de esa manera se vuelve interesante explorar una extensión de este modelo a multiples salidas. Por tal motivo, en este trabajo se presenta una extension de un Proceso Gausiano jerÃąrquico a multiples salidas, usando funciones de covarianza existentes con el objetivo de hacer interpolación y síntesis de movimiento humano. El modelo fue probado con datos tanto artificiales cómo reales, los resultados muestran que el modelo es exitoso interpolando y sintetizando movimiento humano en comparación a un modelo de procesos Gausianos de multiples salidas simple el cuál se usa en este trabajo como referencia.

1 Introduction

Hierarchical modelling has played a major role in several statistical applications. Usually this kind of modelling is related with multiparameter applications, in which there is an assumption that these parameters are related in some way given the structure of the phenomena. Thus, common hierarchical modelling involves putting a prior distribution over these parameters, so a joint probability model will reflect their dependence through the observation of data. However, hierarchical modelling is not only about putting prior distributions over the parameters that govern the observable data, it is common also to express the parameters conditionally in terms of further parameters, known as hyperparameters, and so on [2].

Gaussian Processes (GP) have also been used in a hierarchical way in different kinds of applications. For example in [3] a hierarchical approach is proposed where a mixture of GPs combined with prior assumptions over the parameters is implemented in order to handle the heterogeneity among different realizations of a phenomena, for the case the standing-up movement of paraplegia patients. The hierarchy is constructed when an unobservable latent indicator variable is introduced to determine which is the component of the mixture that will describe a standing-up realization. However, the hierarchical modelling with GPs varies a lot depending on the kind of problem to be solved. Recently another technique using this kind of modelling was introduced in [1], in which a hierarchical single output Gaussian Process (HGP) has been proposed for gene expression time series. This leads to a model with a novel covariance function and suitable for exploiting the relationships between gene replicates. Thus it is capable of making inferences about the underlying and common tendencies followed by these replicates, assumed to be generated from a similar origin. The hierarchy is extendable to several layers, and with some efficient computations it can be extended for clustering of gene-expression time-series data. Other examples of GPs implemented with a hierarchical structure can be found in [4, 5].

In particular the method implemented in [1], mentioned above, proved to be a very reliable model for gene-replicate time-series analysis. However, the model has not been extended to a multiple output framework to exploit the capabilities of coregionalization models. And specifically in applications for Human Motion Synthesis, a hierarchical model with these features has not been explored. The problem of synthesizing realistic human motion is an important issue mainly in the computer animation and game industry, where the demanding of more realistic and human-like characters is increasing. Thus, several techniques have been developed in order to tackle this problem. These techniques can be divided in four categories: manual methods, video-based methods, physics-based methods and motion-capture data-driven methods [6]. Each one of these methods differs in complexity, animator involvement, speed and accuracy. With GPs several approaches have also been developed to tackle this problem also.

In [7] for example, a semiparametric Latent Factor Model is used in order to model the correlations between the degree of freedom of each one of the joints of the skeleton. This implementation was made with the objective of modelling motion variation between several subjects, being able to synthesize different crowd behaviours. Another example using GPs is given in [8] where a hierarchical Gaussian Process Latent Variable Model is introduced. A probabilistic generalization of the Principal Component Analysis (PCA) is made hierarchically by putting a GP prior distribution directly over the latent space. This leads to a model capable of managing several hierarchical extensions for Human Motion Capture data, from a skeleton-wise hierarchy to expressing the interaction of several subjects. In [9–14] several implementations of GPs related with human motion synthesis applications are proposed.

Thus, given that GPs have proven to be a simple and a powerful tool in a range of areas, the main motivation of this work is given by the hypothesis in which the hierarchical assumptions made in [1] related with the common origin associated with a group of gene replicates, can also be made for motion data: there might be an underlying and common trend between different styles of the same kind of motion (walking), and that common trend might transform into a generative model to new motion styles. So, given this single output hierarchical model, it might be possible to extend this particular model to use multiple-output covariances [15] to further applications on several vector-valued regression problems, and specifically in the synthesis of realistic human motion learned from motion-capture data, which is the main purpose of the model proposed in this work.

2 Objectives

2.1 General Objective

To develop a hierarchical multi-output Gaussian process model for interpolation and synthesis of human-like motion data.

2.2 Specific Objectives

1. To formulate a mathematical model for a hierarchical Gaussian process by using covariance functions for vector-valued data.

2. To derive a statistical inference procedure for parameter estimation over this proposed multi-output Gaussian process model.

3. To validate the performance of the multi-output Gaussian process model proposed for interpolation and synthesis in human-like motion data.

3 Background

In this section the techniques used as basis to develop the model proposed in this work are presented. Namely the classic Gaussian Process regression and the hierarchical model developed in [1]. Also, the theory for learning vector valued functions (Multi-output Learning) and some concepts related with computer animation and the specific problems tackled in this work is presented.

3.1 Hierarchical Modelling for Gaussian Processes

3.1.1 Single output Gaussian Process Regression

Gaussian Process Regression is a non-parametric method and assumes a Gaussian Distribution over functions, starting with a distribution for the behaviour we want to identify which can be fully specified by a mean function $m(t)$ (usually zero) and a covariance function $k(t, t')$, and has the following notation:

$$f(t) \sim \mathcal{GP}(m(t), k(t, t')). \tag{1}$$

Thus, the regression process can be summarized as follows: given some observations (noise free) $\mathbf{f}$ of a function $f(t)$ at times $\mathbf{t}$, one might predict the values $\mathbf{f}^*$ at times $\mathbf{t}^*$, then the joint probability distribution of $\mathbf{f}$ and $\mathbf{f}^*$ is given by the next expression:

$$p\left(\begin{bmatrix}\mathbf{f}\\ \mathbf{f}^*\end{bmatrix}\right) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix}\mathbf{K}_{\mathbf{t},\mathbf{t}} & \mathbf{K}_{\mathbf{t},\mathbf{t}^*}\\ \mathbf{K}_{\mathbf{t}^*,\mathbf{t}} & \mathbf{K}_{\mathbf{t}^*,\mathbf{t}^*}\end{bmatrix}\right), \tag{2}$$

where the element (i, j) of the covariance matrix $\mathbf{K}_{\mathbf{t},\mathbf{t}}$ is given by the covariance function $k(t, t')$ evaluated in $\mathbf{t}[i]$ and $\mathbf{t}[j]$; $\mathbf{t}$ and $\mathbf{t}^*$ denotes respectively the time points where the observations are made and where the prediction is desired. After this, by the conditional property the distribution of the desired values conditioned to the observed data is given by,

$$p(\mathbf{f}^* \mid \mathbf{f}) = \mathcal{N}(\mathbf{f}^* \mid \mathbf{K}_{\mathbf{t}^*,\mathbf{t}}\mathbf{K}_{\mathbf{t},\mathbf{t}}^{-1}\mathbf{f}, \mathbf{K}_{\mathbf{t}^*,\mathbf{t}^*} - \mathbf{K}_{\mathbf{t}^*,\mathbf{t}}\mathbf{K}_{\mathbf{t},\mathbf{t}}^{-1}\mathbf{K}_{\mathbf{t},\mathbf{t}^*}). \tag{3}$$

In problems involving real data, usually observations $\mathbf{f}$ are corrupted by some noise, these obervations can be referred as $\mathbf{y}$, in this case and assuming a Gaussian noise the likelihood is given by $p(\mathbf{y} \mid \mathbf{f}) = \mathcal{N}(\mathbf{y} \mid \mathbf{f}, \beta\mathbf{I})$, where β refers to the noise variance and $\mathbf{I}$ represents an identity matrix. With this representation the concept of marginal likelihood can be introduced, which is the integral of the likelihood defined before times the prior defined in equation (1):

$$p(\mathbf{y} \mid \mathbf{t}) = \int p(\mathbf{y} \mid \mathbf{f}, \mathbf{t})p(\mathbf{f} \mid \mathbf{t})d\mathbf{f}. \tag{4}$$

In this case the marginal likelihood is specified in terms of the noise observed values marginalizing variable $\mathbf{f}$. With noise observed values one can obtain a similar representation of the full joint distribution in equation (2) by only adding the noise term $\beta\mathbf{I}$ to $\mathbf{K}_{\mathbf{t},\mathbf{t}}$. After this, the regression process is similar using the posterior equation (3).

3.1.2 The hierarchical model

Several hierarchical models have been proposed to work with statistical applications and Gaussian Processes [3,8,16]. Specifically a powerful and simple hierarchical modelling with a novel covariance function for GPs was introduced by Hensman et al. [1]. The model was motivated first by the application in gene expression time series analysis, and second by the fact that existing models for that specific task did not take into account the potential correlations between biological replicates. These correlations are vital to the structure of the gene expression data.

The key idea about the HGP can be summarized as follows: suppose an unknown parent signal we wish to identify call it n, this parent signal is assumed to be drawn from a zero mean GP $g_n(t)$ with covariance function $k_g(t,t')$. Then, assume one observes child signals from that parent, where each version is assumed to be drawn from a GP denoted by $f_{nr}(t)$ with covariance function $k_f(t,t')$ and is presumed to have as mean $g_n(t)$, thus

$$\begin{aligned} g_n(t) &\sim \mathcal{GP}(0, k_g(t,t')), \\ f_{nr}(t) &\sim \mathcal{GP}(g_n(t), k_f(t,t')). \end{aligned} \tag{5}$$

An example of some samples from this generative model are illustrated in figure 1a. Given a set of N_n observed child signals $\mathbf{Y}_n = \{\mathbf{y}_{nr}\}_{r=1}^{N_n}$ measured at times $\mathbf{T}_n = \{\mathbf{t}_{nr}\}_{r=1}^{N_n}$, where n makes reference to the nth parent and r makes reference to the rth child, the likelihood of all data $\mathbf{Y}_n$ can be written as follows:

$$p(\mathbf{Y}_n \mid \mathbf{T}_n, \boldsymbol{\phi}) = \mathcal{N}(\hat{\mathbf{y}}_n \mid \mathbf{0}, \boldsymbol{\Sigma}_n), \tag{6}$$

where $\hat{\mathbf{y}}_n$ is a column vector used to represent the concatenation of all the elements of $\mathbf{Y}_n$ as $\hat{\mathbf{y}}_n = [\mathbf{y}_{n,1}^\top \mathbf{y}_{n,2}^\top \dots \mathbf{y}_{n,N_n}^\top]^\top$ and $\boldsymbol{\phi}$ represents the parameters of the covariances functions k_g and k_f. The interesting part of the hierarchical model is its simplicity, for example, in the big covariance matrix $\boldsymbol{\Sigma}_n$ (see figure 1b) the corresponding block of $\mathbf{y}_{nr}, \mathbf{y}_{nr'}$ has the following structure:

$$\boldsymbol{\Sigma}_n[r,r'] = \begin{cases} \mathbf{K}_g(\mathbf{t}_{nr}, \mathbf{t}_{nr'}) + \mathbf{K}_f(\mathbf{t}_{nr}, \mathbf{t}_{nr'}) + \beta\mathbf{I} & \text{if r = r'} \\ \mathbf{K}_g(\mathbf{t}_{nr}, \mathbf{t}_{nr'}) & \text{otherwise.} \end{cases}$$

It can be shown why two points in the same same "child signal" ($r = r'$) are jointly Gaussian distributed with zero mean an covariance $k_g(t,t') + k_f(t,t')$. It is also straightforward to check that two points placed in differents "child signals" have covariance $k_g(t,t')$ (See Appendix A.1). Thus, if inference about functions $f_{nr}(t)$ or $g_n(t)$ is also needed, then, the covariance between the data and the functions is neccesary, which can be defined as:

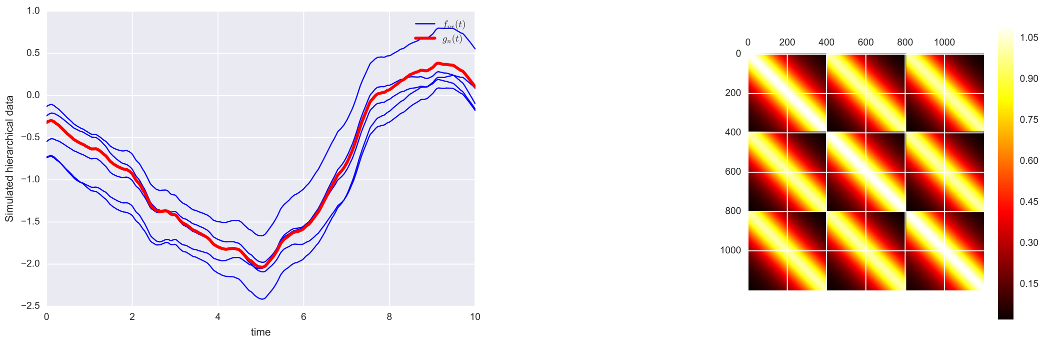
$$\mathrm{cov}(\mathbf{y}_{n,r}^{(t)}, g_n(t')) = k_g(t,t') \tag{7}$$

$$\mathrm{cov}(\mathbf{y}_{n,r}^{(t)}, f_{nr'}(t')) = \begin{cases} k_g(t,t') + k_f(t,t') & \text{if r = r'} \\ k_g(t,t') & \text{otherwise,} \end{cases}$$

where the superscripted $\mathbf{y}_{n,r}^{(t)}$ denote the element of $\mathbf{y}_{n,r}$ observed at time t. The method used to make inference is similar to the one used in the single output regression, the optimization of the hyper-parameters can be made using standard methods such as type II Maximum Likelihood. This model can be extended also to deeper hierarchies, for example a three layer hierarchy model can be specified as:

$$\begin{aligned} g_n(t) &\sim \mathcal{GP}(0, k_g(t,t')), \\ e_{ni}(t) &\sim \mathcal{GP}(g_n(t), k_e(t,t')) \\ f_{nir}(t) &\sim \mathcal{GP}(e_{ni}(t), k_f(t,t')). \end{aligned} \tag{8}$$

Call $g_n(t)$ the underlying trend, $e_{ni}(t)$ the ith parent and $f_{nir}(t)$ the rth children from the ith parent. Similar properties as the two layer hierarchy are fulfilled, for example two observed points on one of the replicates in the last layer of the hierarchy will be jointly Gaussian distributed with zero mean, but now with covariance $k_g(t,t') + k_e(t,t') + k_f(t,t')$. This shows that the model remains simple while adding more layers to the hierarchy.



(a) (b)

Figure 1: Single Output Hierarchical model illustration. (a) Samples of the generative process in equation 5. The red signal is the sample from the parent process. (b) Likelihood covariance matrix of the single output hierarchical model. This is the matrix defined in equation 3.1.2.

3.2 Linear Model of Corregionalization(LMC)

The LMC or Linear Model of Corregionalization [15] has been widely used in multioutput learning and vector-valued prediction. The LMC was developed in order to construct valid covariance functions to generalize the concept of covariance from the single output to the multiple output case. The LMC assumes that each output is represented as a linear combination of Q latent functions, so for a set of D outputs $\{f_d(\mathbf{t})\}_{d=1}^{D}$ of a vector valued function $\mathbf{f}(\mathbf{t})$ where $\mathbf{t} \in \mathbf{R}^p$, the representation of each output is given by:

$$f_d(\mathbf{t}) = \sum_{q=1}^{Q} a_{d,q} u_q(\mathbf{t}), \tag{9}$$

where $a_{d,q}$ are coefficients and each latent function $u_q(\mathbf{t})$ is generated independently, has zero mean and covariance $\mathrm{cov}[u_q(\mathbf{t}), u_{q'}(\mathbf{t})] = k_q(\mathbf{t}, \mathbf{t}')$ only when $q = q'$. Despite the processes $\{u_q(\mathbf{t})\}_{q=1}^{Q}$ are independent, they can have the same covariance function $k_q(\mathbf{t}, \mathbf{t}')$ when $q \neq q'$. Thus a similar expression for $\{f_d(\mathbf{x})\}_{d=1}^{D}$ can be written grouping the functions $u_q(\mathbf{t})$ that share the same covariance as:

$$f_d(\mathbf{t}) = \sum_{q=1}^{Q} \sum_{i=1}^{R_q} a_{d,q}^{i} {u^i}_q(\mathbf{t}), \tag{10}$$

this expression means that there are Q groups of functions ${u^i}_q(\mathbf{t})$ and that each one of this functions that are on the same group share the same covariance, but are independent. The multioutput modelling also proposes to establish the correlation between the outputs through $\mathrm{cov}[f_d(\mathbf{t}), f_{d'}(\mathbf{t}')] = (\mathbf{K}(\mathbf{t}, \mathbf{t}'))_{d,d'}$, which is given in terms of the latent functions ${u^i}_q(\mathbf{t})$. So given that the functions ${u^i}_q(\mathbf{t})$ are independent one can express that covariance as follows:

$$(\mathbf{K}(\mathbf{t}, \mathbf{t}'))_{d,d'} = \sum_{q=1}^{Q} \sum_{i=1}^{R_q} a_{d,q}^{i} a_{d',q}^{i} k_q(\mathbf{t}, \mathbf{t}'), \tag{11}$$

Then if we group all the coefficients $b_{d,d'}^{q} = \sum_{i=1}^{R_q} a_{d,q}^{i} a_{d',q}^{i}$, the kernel $\mathbf{K}(\mathbf{t}, \mathbf{t}')$ can now be expressed as:

$$\mathbf{K}(\mathbf{t}, \mathbf{t}') = \sum_{q=1}^{Q} \mathbf{B}_q k_q(\mathbf{t}, \mathbf{t}'), \tag{12}$$

where $\mathbf{B}_q \in \mathbf{R}^{D \times D}$ refers to the coregionalization matrix associated with the latent function q containing the coefficients $b_{d,d'}^{q}$. This matrix encodes the covariance between the D outputs. The rank of each matrix $\mathbf{B}_q$ is determined by the coefficient R_q. Another type of modelling that is a special case of LMC is the Intrinsic Corregionalization Model (ICM), which assumes that each output f_d is only approximated with one latent function, so $Q = 1$, and assuming a complete dataset $\mathbf{T}$ the kernel matrix takes the form:

$$\mathbf{K}(\mathbf{T},\mathbf{T}) = \mathbf{B} \otimes k(\mathbf{T},\mathbf{T}), \tag{13}$$

where $\otimes$ is the Kronecker product between matrices. The advantage of using this coregionalization model lies in its simplicity, the inference algorithm for this model is mathematically easier to derive essentially for the properties of the Kronecker product. The covariance for the input space $k(\mathbf{T},\mathbf{T})$ can be any valid covariance function to produce positive semidefinite matrices, as the Squared Exponential kernel. The LMC can be represented in a similar way for a complete data set $\mathbf{T}$ like equation (13) by only adding more coregionalization matrices as:

$$\mathbf{K}(\mathbf{T},\mathbf{T}) = \sum_{q=1}^{Q} \mathbf{B}_q \otimes k(\mathbf{T},\mathbf{T}). \tag{14}$$

3.2.1 Corregionalization Matrix

It is worth to mention that the coregionalization matrix $\mathbf{B}_q$ referenced before is directly estimated from the observed data. Thus, in order to meet the conditions of positive-semidefiniteness, this matrix has to be parameterized in a way that ensures this constraint. This parametrization has the following form:

$$\mathbf{B}_q = \mathbf{W}_q \mathbf{W}_q^\top + diag(\boldsymbol{\kappa}_q), \tag{15}$$

where $\mathbf{W}_q$ is a $D \times R_q$ matrix and $\boldsymbol{\kappa}_q \in \mathbf{R}^D$. This representation allows to have a low rank form of $\mathbf{B}_q$, an also allows to reduce the amount of parameters to be estimated. The vector $\boldsymbol{\kappa}_q$ is added to allow each one the outputs to have and independent behaviour.

3.3 Computer-based human animation with motion capture

Computer-based human animation is a major area of research in computer graphics. The main idea is to develop techniques, algorithms or methodologies capable of generating human motion in a realistic manner. One specific and widely used methodology is the motion capture technique in which human motion is given by physical models that describe the motion in a mathematical way, or with motion-capture special equipment in which a set of sensors is placed on a person who then performs different kinds of movements in order to get that data in a 3D representation by specialized software [6].

This technique is widely used by top companies such as Disney, Pixar, Dreamworks or EA, which develop many graphics applications for the computer game and animation filming industry. Human motion capture data can be used in different manners, for example, it is not only used to synthesize purely human motion but also to create new ways of animating non-human animated characters, such as animals [17], or to teach a robot how to behave like a human when it is performing a movement [18].

Diving more into motion-capture, the set of sensors on the performer's body has very specific locations, see figure 2. The arrangement of those sensors can be described as a skeleton with a specific hierarchy. Notice that the sensors are located on several key joints of the body; in this way the specialists can capture the major activity of the human body while the motion is being performed. After the data given by the sensors is further processed, it can be stored in several kinds of formats ready for any kind of processing needed. Visit the CMU Graphics Lab Motion Capture Database for more details [1].

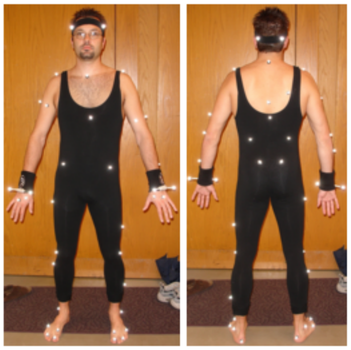


Figure 2: Example of arrangement of sensors on performers body. Taken from MOCAP official website.

3.4 Human Motion Synthesis and Interpolation

Human Motion Synthesis (HMS) refers to several computer-based techniques used to perceive and generate motion performed by humans. These kind of techniques are used in computer animation, simulators, computer and console game technology and robotics. HMS has become an interesting area of research, with impact not only in academy research but also in industry. There are several approaches to generate Human Motion in a realistic way: manual methods, video-based methods, physics-based methods and motion-capture data-driven methods [6].

Manual methods consist in manually setting the joints in skeleton on each frame before generating the motion sequence using specialized algorithms. Besides, these methods are characterized by an expert in animation of motion figures that has to be present to help the process of synthesis to be more accurate. Some examples of this kind of approach can be found in [19, 20]. Video-based methods usually have more elements to consider given that they use computer vision techniques and several cameras to track and describe human motion in a video obtaining 3D information of the scene. For example, in [21] they combine image reconstruction from video techniques, to recreate 3D characters in a virtual environment to further control and synthesize different kinds of motions from different views of people. Another examples of these methodologies can be found in [22–26].

[1] The CMU Graphics Lab Motion Capture Database was created with funding from NSF EIA-0196217 and is available at http://mocap.cs.cmu.edu

Physics-based methods are amongst the most elaborated methods, because they require the formulation of dynamic models through differential equations to describe motion according to physical constraints. Usually these differential equations are made to describe the trajectories of the joints of the body after they are properly solved. For example in [27] an animation system is constructed based on a set of objective functions to be optimized that eventually lead to linear time analytical first derivatives. The system is robust because the set of functions properly includes physical constraints given by the animator such as ground contact, in that way it is capable of synthesizing several kinds of motions with a high level of dynamics. Similar approaches use optimization to meet user constraints and synthesize motion. However, there are other physically inspired techniques that use control theory. For example in [28] the authors proposed a system to animate athletes in which the principles of control theory were applied in order to give the characters the ability to maintain balance while moving and to simulate complicated kinds of motions meeting the specific physical constraints associated with the athletes. Some other examples of physics-based motion synthesis can be found in [29–32].

Finally, motion-capture data-driven methods refer to the reuse of existing motion data, which can be generated by one of the methods explained above or generated by special motion-capture equipment (See Fig. 2). Specially, data-driven methods have been tackled in two major ways, graph-based techniques and statistical motion synthesis techniques [6]. The graph-based techniques [33–36] almost always have the same path, first taking as input the motion data generated by the special equipment, then trying to find similarities between those motions and setting transitions between similar motions. Next, the construction of the graph takes place given those transitions, and finally paths from this graph are chosen in order to generate new motions satisfying some predefined conditions of optimality. On the other hand, statistical motion synthesis techniques are not as easy to generalize as the graph-based methods, given the fact that several approaches can be taken in order to sythesize motion data in this way.

Statistical data-driven models for HMS have been widely used because of their generalization capacity, and because these methods are simpler mathematically than other widely used methods, like the physics-based. These methods usually differ in the process depending on the probabilistical model used to learn or explain the motion data.

One common statistical approach is to use state-space models like Markov models. For example, Brand et al. [37] used a Hidden Markov Model (HMM) in order to develop a model of structure/style from a motion sequence combined with a multidimensional style variable $\boldsymbol{v}$ in order to take a training set of motions, then learn the parameters of the model, and thus, it was capable of generating new and stylistic motions. Here, stylistic motions refer to motion samples that follow the same kind of movement, such as walking, but with different behaviour. Another approach [38] used a Markov Chain plus an HMM in two levels. In the first level the Markov chain is used to model the transition between joints and in the second level the HMM is implemented to generate motion segments. Another example of Markov models was introduced by Wang et al. [39], where a Markov chain was used to switch between Linear Dynamical Systems to generate new motions.

Latent space models have also been used to reduce the dimensionality and create simpler models to synthesize human motion. In [8] a generalization of PCA was used to apply it to human motion data, mixed with Gaussian Process priors over the mapping to the latent space. This resulted in a hierarchical and non-parametric model, including the possibility to model the interaction between different motion subjects. Other ways of using Gaussian process and latent variable models have also been proposed [40,41]. The first tries to estimate a PDF of the latent states to sample different kinds of sequences. The second one is a state-of-the-art model that uses multivariate GPs combined with a semi-parametric Latent Factor model to synthesize motion variation in several subjects. Other examples of different statistical data-driven approaches can be found in [42–46].

Other more versatile techniques have been developed that involve other kinds of approaches. In [47] non-linear models of deformation are used to learn the shape and deformation of an object, being capable of making an extension to motion synthesis through shape and style information gathered from data. Another way is to use hybrid methods to create more robust approaches gathering the strengths of two or several methodologies [6]. It is very common to see works in which both the strengths of the motion capture methods and the physics-based methods are applied. Some examples of these ideas can be found in [48–53].

The human motion interpolation (HMI) problem is often related with marker-based motion-capture approaches Sometimes the motion capture software misses the trajectory of the subject in short intervals of time. It is often usual to let motion editors complete the sequences given the recorded ones [54]. However several works with motion capture data have focused on developing techniques to properly reconstruct the missing trajectories. For synthesis the interpolation is also commonly used. The process of interpolation can be generalized as: several keyframes are selected for each motion in the database, then these keyframes are used as a base to interpolate the motion between them, and finally some approaches try to introduce some kind of style mechanism in order to interpolate/synthesize stylistic motion.

The interpolation methods differ in the techniques used in each one of the stages presented before. Usually the keyframe selection is made manually. Then, after selecting the keyframes, the interpolation part can be made using geometric techniques [55, 56], linear combination of kernel functions to fill the motion between keyframes [57,58], signal processing [59,60], physical assumptions with control theory [61] and introducing some statistical modelling through multidimensional analysis as in [62, 63]. Some of these approaches usually implement some kind of inverse kinematics in order to properly generate human motion constrained by physical laws.

The style over the interpolated motion can be made in several ways. For example in [63] they propose to estimate a probability density function (PDF) to describe the motions in the training examples, being able to add some randomness at the time of generating motions through interpolation. However the method tends to prefer samples similar to the training samples. In [62] a multidimensional abstract space is used to encode the information of the subjects, and is called control space. The parameters in this control space are called the control parameters. Four of these five parameters control the spatial constraints and the final parameter is used to control the desired style to apply over the interpolated motion. The style over motion usually is related to probabilistic approaches to tackle this problem both in synthesis and interpolation.

4 Materials and Methods

In this section the proposed model is properly explained and derived in the first part, several appendix sections were referenced to help in the understanding of some of the computations involving all aspects of the model. Also, the parameter estimation procedure is further explained and finally some correlation measures are defined in order to quantify the performance of a model to synthesize new subjects.

4.1 Multiple Output Hierarchical Gaussian Process (MOHGP)

The MOHGP is the model proposed in this work, is an extension of the model specified in 3.1.2 proposed by Hensman et al. (2013). The MOHGP model uses the same structure for the hierarchy, however $g_n(t)$ and $f_{nr}(t)$ for the case of the two layer hierarchy are no longer a single output signal but instead vector valued functions called $\mathbf{g}_n(t)$ and $\mathbf{f}_{nr}(t)$ respectively. This change influences in the form of the likelihood function by only introducing a different covariance function for the parent and child process. In this case the covariance functions have to take into account that the observed signals have multiple outputs rather than only one, thus the proposed model can be specified as:

$$\begin{aligned} \mathbf{g}_n(t) &\sim \mathcal{GP}(\mathbf{0}, \mathbf{K}_\mathbf{g}(t,t')), \\ \mathbf{f}_{nr}(t) &\sim \mathcal{GP}(\mathbf{g}_n(t), \mathbf{K}_\mathbf{f}(t,t')), \end{aligned} \tag{16}$$

where,

$$\begin{aligned} \mathbf{K}_\mathbf{g}(t,t') &= \sum_{q=1}^{Q} \mathbf{B}_{gq} k_{gq}(t,t'), \\ \mathbf{K}_\mathbf{f}(t,t') &= \sum_{l=1}^{L} \mathbf{B}_{fl} k_{fl}(t,t'). \end{aligned} \tag{17}$$

The two expressions above correspond to the Linear Model of Corregionalization (LMC) for vector valued functions, previously explained in section 3.2. The Intrinsic Model of Corregionalization is the special case when $q = 1$ and $l = 1$. Even though $q \neq l$, the amount of outputs in each layer must be the same. Here we assume D outputs for the vector valued function $\mathbf{g}_n(t)$. As well as the single output hierarchical model, one can demonstrate that two points in the same child signal (multiple output for this case) are jointly Gaussian distributed with zero mean and covariance $\mathbf{K}_\mathbf{g}(t,t') + \mathbf{K}_\mathbf{f}(t,t')$ (see Appendix A.2). Also, two points in different child signals have covariance $\mathbf{K}_\mathbf{g}(t,t')$.

Given a set of N_n observed child vector valued functions each one denoted as a matrix $\mathbf{Y}_{nr}$ composed of column vectors as $\mathbf{Y}_{nr} = [\mathbf{y}_{nr}^1 \mathbf{y}_{nr}^2 \cdots \mathbf{y}_{nr}^D]$, in which $\mathbf{y}_{nr}^d = \mathbf{y}_{nr}^d + \beta\mathbf{I}$ assuming noisy observations, where where D is the dimension of the function measured at N_t times $\mathbf{t}_{nr}$, $\mathbf{t}_{nr}$ here is the same for each one of the D outputs for simplicity, where n makes reference to the nth parent and r makes reference to the rth child. One can arrange all data in the set $\mathbf{Y}_n$ for an specific parent with all the measurement times $\mathbf{T}_n$, where $\mathbf{Y}_n = \{\mathbf{Y}_{nr}\}_{r=1}^{N_n}$ and $\mathbf{T}_n = \{\mathbf{t}_{nr}\}_{r=1}^{N_n}$ which can be different measurement time points for different replicates, leading to write all the data $\mathbf{Y}_n$ marginal likelihood as follows:

$$p(\mathbf{Y}_n \mid \mathbf{T}_n, \boldsymbol{\phi}) = \mathcal{N}(\hat{\mathbf{y}}_n \mid \mathbf{0}, \boldsymbol{\Sigma}_n), \tag{18}$$

where $\hat{\mathbf{y}}_n$ is a big column vector used to represent the concatenation of all the elements of $\mathbf{Y}_n$ and each output vector of $\mathbf{Y}_{nr}$ as $\hat{\mathbf{y}}_n = [\hat{\mathbf{y}}_{n,1}^\top \hat{\mathbf{y}}_{n,2}^\top \dots \hat{\mathbf{y}}_{n,N_n}^\top]^\top$, here each column vector $\hat{\mathbf{y}}_{n,r} = [\mathbf{y}_{nr}^{1\top} \mathbf{y}_{nr}^{2\top} \dots \mathbf{y}_{nr}^{D\top}]^\top$ represents the concatenation of all the column vectors of the matrix $\mathbf{Y}_{nr}$. The vector $\boldsymbol{\phi}$ represents all the parameters of the covariance functions $\mathbf{K}_\mathbf{g}(\mathbf{t}, \mathbf{t}')$ and $\mathbf{K}_\mathbf{f}(\mathbf{t}, \mathbf{t}')$, these parameters include, the hyperparameters of the input space covariance functions k_{gq} and k_{fl} and the entries of all the corregionalization matrices for the output space covariances $\mathbf{B}_{gq}$ and $\mathbf{B}_{fl}$.

By the properties mentioned before and demonstrated in Appendix A.2, the mean of the likelihood function will be a zero mean vector of size equal to the amount of replicates N_n times the d outputs of each replicate and times the amount of input points for each output. The covariance matrix $\boldsymbol{\Sigma}_n$ will have the following block structure for the replicates $\mathbf{Y}_{nr}$ and $\mathbf{Y}_{nr'}$:

$$\boldsymbol{\Sigma}_n[r, r'] = \begin{cases} \mathbf{K}_\mathbf{g}(\hat{\mathbf{t}}_{nr}, \hat{\mathbf{t}}_{nr'}) + \mathbf{K}_\mathbf{f}(\hat{\mathbf{t}}_{nr}, \hat{\mathbf{t}}_{nr'}) + \boldsymbol{\Gamma} & \text{if r = r'} \\ \mathbf{K}_\mathbf{g}(\hat{\mathbf{t}}_{nr}, \hat{\mathbf{t}}_{nr'}) & \text{otherwise,} \end{cases} \tag{19}$$

where $\boldsymbol{\Gamma} = \mathbf{A} \otimes \mathbf{I}$ is a diagonal matrix where $\mathbf{A} \in \mathbf{R}^{D \times D}$ which elements are $\{\beta_d\}_{d=1}^{D}$ the noise variances for each one of the outputs, $I \in \mathbf{R}^{N_t \times N_t}$ and $\hat{\mathbf{t}}_{nr} = [\mathbf{t}_{nr}^{1}{}^\top \mathbf{t}_{nr}^{2}{}^\top \dots \mathbf{t}_{nr}^{D}{}^\top]^\top$ the stacked column vector of the inputs for each one of the D outputs of the rth child, here $\mathbf{t}_{nr}^{d} = \mathbf{t}_{nr}^{d'}$ for simplicity. The block covariance in equation (19) is very similar to the one of the single output model, however the block $[r, r']$ are different because of the introduction of the multiple output covariances. In figure 3 there is a graphical explanation of the proposed model ilustrating the shape of the covariance matrix of the model likelihood. If inference about functions $\mathbf{f}_{nr}(t)$ (some children) or $\mathbf{g}_n(t)$ (underlying trend) is needed, then, the covariance between the data and the functions is necessary which is the same as in the single output case and is defined as:

$$\text{cov}(\mathbf{Y}_{n,r}^{(t)}, \mathbf{g}_n(t')) = \mathbf{K}_\mathbf{g}(t, t') \tag{20}$$

$$\text{cov}(\mathbf{Y}_{n,r}^{(t)}, \mathbf{f}_{nr'}(t')) = \begin{cases} \mathbf{K}_\mathbf{g}(t, t') + \mathbf{K}_\mathbf{f}(t, t') & \text{if r = r'} \\ \mathbf{K}_\mathbf{g}(t, t') & \text{otherwise,} \end{cases} \tag{21}$$

where the superscripted $\mathbf{Y}_{n,r}^{(t)}{}^\top$ denotes a column vector extracted from $\mathbf{Y}_{n,r}$. This vector represents the value of each one of the outputs (column vectors) of $\mathbf{Y}_{n,r}$ at time t. Inference over the parent $\mathbf{g}_n(t)$ given the observed children $\mathbf{Y}_n$ is similar to the one used in single output regression explained in section 3.1.1, here can be summarized in equation as follows:

$$p(\hat{\mathbf{g}}_n \mid \hat{\mathbf{y}}_n) = \mathcal{N}(\hat{\mathbf{g}}_n \mid \mathbf{K}_{\mathbf{t}^*,\mathbf{t}}\mathbf{K}_{\mathbf{t},\mathbf{t}}^{-1}\hat{\mathbf{y}}_n, \mathbf{K}_{\mathbf{t}^*,\mathbf{t}^*} - \mathbf{K}_{\mathbf{t}^*,\mathbf{t}}\mathbf{K}_{\mathbf{t},\mathbf{t}}^{-1}\mathbf{K}_{\mathbf{t},\mathbf{t}^*}), \tag{22}$$

where $\mathbf{t}^*$ makes reference to the inputs in which the vector valued function will be inferred and $\mathbf{t}$ is equal to the set of all the inputs $\mathbf{T}_n = \{\mathbf{t}_{nr}\}_{r=1}^{N_n}$ of the observed children $\mathbf{Y}_n$. The same equation (22) can be applied to make inference over one of the children in $\mathbf{Y}_n$. Thus, given the rules defined in equation (20) is straightforward to derive that:

$$\begin{aligned}\mathbf{K}_{\mathbf{t}^*,\mathbf{t}}^{\top} &= \mathbf{K}_{\mathbf{t},\mathbf{t}^*} = \mathbf{K}_{\mathbf{g}}(\mathbf{t}, \mathbf{t}^*),\\ \mathbf{K}_{\mathbf{t}^*,\mathbf{t}^*}^{\top} &= \mathbf{K}_{\mathbf{g}}(\mathbf{t}^*, \mathbf{t}^*),\\ \mathbf{K}_{\mathbf{t},\mathbf{t}} &= \mathbf{\Sigma}_n \text{ Marginal likelihood matrix.}\end{aligned}$$

The optimization of the hyper-parameters can be made using standard methods such as type II Maximum Likelihood. For this purpose it is neccesary to extract the likelihood gradients with respect to the model parameters. In the following section this process is further explained.

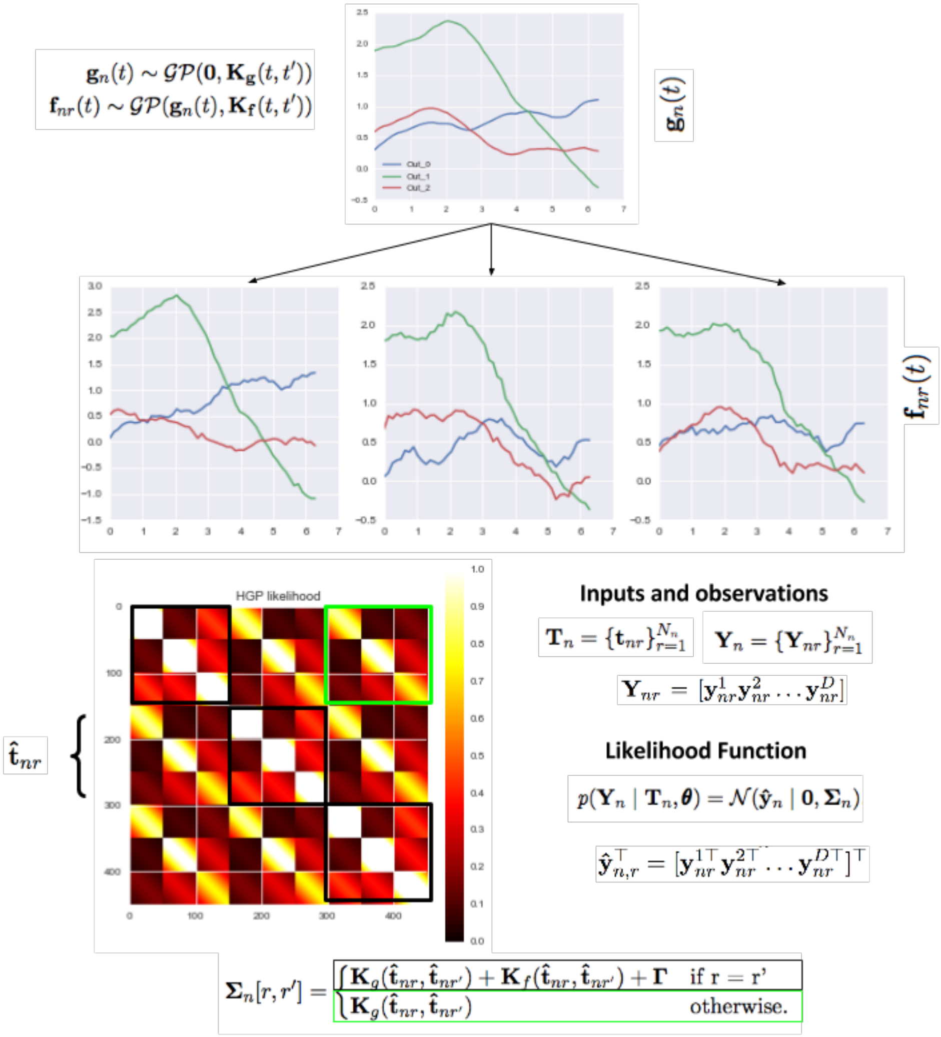

$\mathbf{g}_n(t) \sim \mathcal{GP}(\mathbf{0}, \mathbf{K_g}(t,t'))$
$\mathbf{f}_{nr}(t) \sim \mathcal{GP}(\mathbf{g}_n(t), \mathbf{K_f}(t,t'))$
$\mathbf{g}_n(t)$
Out_0
Out_1
Out_2
$\mathbf{f}_{nr}(t)$
HGP likelihood
$\hat{\mathbf{t}}_{nr}$
Inputs and observations
$\mathbf{T}_n = \{\mathbf{t}_{nr}\}_{r=1}^{N_n}$
$\mathbf{Y}_n = \{\mathbf{Y}_{nr}\}_{r=1}^{N_n}$
$\mathbf{Y}_{nr} = [\mathbf{y}_{nr}^1 \mathbf{y}_{nr}^2 \cdots \mathbf{y}_{nr}^D]$
Likelihood Function
$p(\mathbf{Y}_n \mid \mathbf{T}_n, \boldsymbol{\theta}) = \mathcal{N}(\hat{\mathbf{y}}_n \mid \mathbf{0}, \boldsymbol{\Sigma}_n)$
$\hat{\mathbf{y}}_{n,r}^\top = [\mathbf{y}_{nr}^{1\top} \mathbf{y}_{nr}^{2\top} \cdots \mathbf{y}_{nr}^{D\top}]^\top$
$\boldsymbol{\Sigma}_n[r,r'] = \begin{cases} \mathbf{K}_g(\hat{\mathbf{t}}_{nr}, \hat{\mathbf{t}}_{nr'}) + \mathbf{K}_f(\hat{\mathbf{t}}_{nr}, \hat{\mathbf{t}}_{nr'}) + \boldsymbol{\Gamma} & \text{if r = r'} \\ \mathbf{K}_g(\hat{\mathbf{t}}_{nr}, \hat{\mathbf{t}}_{nr'}) & \text{otherwise.} \end{cases}$

Figure 3: **Ilustration of a Multiple Output Hierarchical GP based on [1].** Top: The vector valued funcion $\mathbf{g_n}(t)$ is the underlying trend(parent) followed by the $\mathbf{f}_{nr}(t)$ replicate (children) sample versions deviating with a small variance and some noise. Bottom left: The marginal likelihood matrix $\boldsymbol{\Sigma}_n$, the diagonal blocks are related with each children sample. For the case the covariances $\mathbf{K_g}$ and $\mathbf{K_f}$ come from the Intrinsic Corregionalization Model (ICM). Standard methods can be used for optimize covariance hyperparameters. The same equation for classic GP prediction can be used here to determine the underlying trend $\mathbf{g_n}(t)$ given the children observations. Also new children prediction can be made in the same way.

4.2 Parameter Estimation Procedure: MOHGP

For the proposed model the same standard methods used for parameter estimation of Gaussian Process can also be used. In this case the form of the likelihood function resembles the one a simple multiple output model in the sense that the diagonal of the matrix $\mathbf{\Sigma}_n$ is the covariance of each one of the observed children. Thus, taking as reference the method developed in [15], the process for estimating the parameters of the proposed model is derived as follows:

Our goal is to estimate the parameters $\boldsymbol{\phi}$ that maximize the marginal likelihood function, or in other words, the parameters that best fit our observed values $\mathbf{Y}_n$. In that sense the objective function here is the log marginal likelihood

$$\log p(\mathbf{Y}_n \mid \mathbf{T}_n, \boldsymbol{\phi}) = \log \mathcal{N}(\hat{\mathbf{y}}_n \mid \mathbf{0}, \mathbf{\Sigma}_n),$$

which is straightforwardly derived as

$$\log p(\mathbf{Y}_n \mid \mathbf{T}_n, \boldsymbol{\phi}) = -\frac{1}{2}\hat{\mathbf{y}}_n^\top \mathbf{\Sigma}_n^{-1} \hat{\mathbf{y}}_n - \frac{1}{2} \log \mid \mathbf{\Sigma}_n \mid - \frac{N_n N_t D}{2} \log(2\pi), \tag{23}$$

where N_n, N_t, and D makes reference respectively, to the amount of observed chilren, the amount of input points per output and the amount of outputs per child. To maximize this marginal likelihood function it is neccesary to derive the gradients to implement a numerical optimization strategy. The derivatives with respect to one of the parameters ϕ_i inside the parameter vector $\boldsymbol{\phi}$ can be computed as

$$\frac{\partial \log p(\mathbf{Y}_n \mid \mathbf{T}_n, \boldsymbol{\phi})}{\partial \phi_i} = -\frac{1}{2}\hat{\mathbf{y}}_n^\top \frac{\partial \mathbf{\Sigma}_n^{-1}}{\partial \phi_i} \hat{\mathbf{y}}_n - \frac{1}{2} \frac{\partial \log \mid \mathbf{\Sigma}_n \mid}{\partial \phi_i}, \tag{24}$$

By the properties of the derivatives of matrices respect to scalars the derivative of the marginal likelihood with respect to one of the hyperparameters is given by

$$\frac{\partial \log p(\mathbf{Y}_n \mid \mathbf{T}_n, \boldsymbol{\phi})}{\partial \phi_i} = -\frac{1}{2}\hat{\mathbf{y}}_n^\top \mathbf{\Sigma}_n^{-1} \frac{\partial \mathbf{\Sigma}_n}{\partial \phi_i} \mathbf{\Sigma}_n^{-1} \hat{\mathbf{y}}_n - \frac{1}{2} \operatorname{tr}\left(\mathbf{\Sigma}_n^{-1} \frac{\partial \mathbf{\Sigma}_n}{\partial \phi_i} \right). \tag{25}$$

After computing de derivative $\frac{\partial \mathbf{\Sigma}_n}{\partial \phi_i}$ (See appendix A.3) for all the model parameters, the gradient vector can be used in a gradient based optimization algorithm.

4.3 Synthesis Performance Measures

The synthesis process is the process in which new samples of human motion subjects usually are generated using some model. Usually this kind of process is evaluated in a visual manner, sometimes with the help of an expert in computer generated animation. However, for the purpose of making a more quantifiable comparison between models for synthesizing motion data, we considered two correlation measures to determine if an artificially generated subject exhibits a similar behaviour to a real one.

Synthesis Evaluation To evaluate the ability of a model to generate new samples $\mathbf{Y}_s$ similar to real observations $\mathbf{Y}_n$ it was necessary to use a correlation measure between the samples generated by the trained models and the real ones, which are different from the training set. Here $\mathbf{Y}_s = \{\mathbf{Y}_{sr}\}_{r=1}^{N_s}$ and $\mathbf{Y}_n = \{\mathbf{Y}_{nr}\}_{r=1}^{N_n}$, where $\mathbf{Y}_{sr} = [\mathbf{y}_{sr}^1 \mathbf{y}_{sr}^2 ... \mathbf{y}_{sr}^D]$ and $\mathbf{Y}_{nr} = [\mathbf{y}_{nr}^1 \mathbf{y}_{nr}^2 ... \mathbf{y}_{nr}^D]$. In this case the two correlation measures used were:

1. Discrete Cross Correlation, which is defined as:

$$(\mathbf{y}_{nr}^d \star \mathbf{y}_{sr}^d)(n) = \sum_{-\infty}^{\infty} \mathbf{y}_{nr}^d[m]\mathbf{y}_{sr}^d[m+n],$$

   where $\mathbf{y}_{nr}^d[m]$ is one of the outputs of a real observation at time index m and $\mathbf{y}_{sr}^d[m+n]$ is the generated sample at index time $m+n$, here n is the lag between the signals.

2. Pearson Linear Correlation, which is defined as:

$$\rho_{\mathbf{y}_{nr}^d, \mathbf{y}_{sr}^d} = \frac{\mathrm{cov}(\mathbf{y}_{nr}^d, \mathbf{y}_{sr}^d)}{\sigma_{\mathbf{y}_{nr}^d} \sigma_{\mathbf{y}_{sr}^d}},$$

   where σ makes reference to the standard deviation of each variable. It is important to remark here that it is possible that the generated subjects to start at a different position from the real ones. Thus, the cross correlation will be more robust to this kind of cases.

For evaluating the performance of synthesis, the following metrics are defined, given the correlation measures defined before:

Mean Cross Correlation (MCC): This metric gives an intuition of the correlation between the generated vector valued signals $\mathbf{Y}_s$ and the real ones $\mathbf{Y}_n$. The greater the value of this metric is, the better the model is to generate samples similar to the real ones. To compute this measure the following procedure is followed:

1. Normalize the data in $\mathbf{Y}_n$ and in $\mathbf{Y}_s$.
2. Take a new sample $\mathbf{y}_{sr}$ from $\mathbf{Y}_s$.
3. Take a new sample $\mathbf{y}_{nr}$ from $\mathbf{Y}_n$.
4. Take the output d from $\mathbf{y}_{nr}$ and the corresponding one from $\mathbf{y}_{sr}$.
5. Compute the cross correlation between $\mathbf{y}_{nr}^d$ and $\mathbf{y}_{sr}^d$ for different values of m and take the maximum value.

6. Repeat from step 5 until $d = D$.
7. Compute the mean cross correlation across all the outputs.
8. Repeat from step 3 until there are no more real observations.
9. Repeat from step 2 until there are no more generated samples.
10. Compute the mean and the standard deviation of the Mean Cross Correlation measures across all the new samples in $\mathbf{Y}_s$.

Mean Linear Correlation (MLC): The linear correlation (Pearson) of a signal with itself is always going to be +1, so one will expect that the linear correlation of the generated samples and the real ones to be very close to +1, so to compute this measure of linear correlation the following process is performed:

1. Normalize the data in $\mathbf{Y}_n$ and in $\mathbf{Y}_s$.
2. Take a new sample $\mathbf{y}_{sr}$ from $\mathbf{Y}_s$.
3. Take a new sample $\mathbf{y}_{nr}$ from $\mathbf{Y}_n$.
4. Take the output d from $\mathbf{y}_{nr}$ and the corresponding one from $\mathbf{y}_{sr}$.
5. Compute the Pearson linear correlation between $\mathbf{y}_{nr}^d$ and $\mathbf{y}_{sr}^d$.
6. Repeat from step 5 until $d = D$.
7. Compute the mean linear correlation across all the outputs.
8. Repeat from step 3 until there are no more real observations.
9. Repeat from step 2 until there are no more generated samples.
10. Compute the mean and the standard deviation of the Mean Linear Correlation measures across all the new samples in $\mathbf{Y}_s$.

5 Experimental Results and Discussion

The proposed model was tested with both artificial and real data. The latter was obtained from MOCAP data set offered by Carnegie Mellon CMU. For the tests the proposed model (MOHGP) was compared with a multiple output GP model (Called MOA here) with no hierarchy structure involved. Two tasks were evaluated, first, interpolation over one of the artificial/MOCAP subjects given the other observed subjects, and second, generation of new artificial/MOCAP samples.

5.1 Interpolation

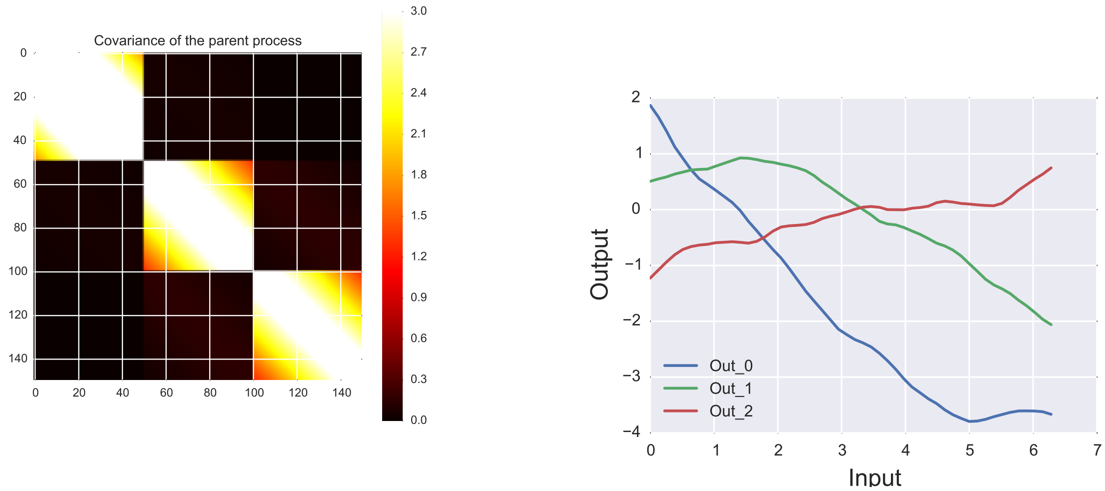
5.1.1 Artificial Data

Two layer hierarchy To perform this test the data was generated using the generative model specified in equation (16). Thus, it was necessary to define two kernels $\mathbf{K_g}$ and $\mathbf{K_f}$, the kernels for the first layer (parent) and the second layer (children), respectively. An LMC was used to represent the covariance function for the GP's at each layer. Here the LMC was assumed to be the sum of several ICM (see eq. (13)), three ICMs were used to define each layer. The parameters of both covariances were randomly generated, nevertheless it was ensured that the variance of the second layer GP was less than the first layer variance. White noise was also added to the second layer covariance function. the white noise was fixed to three different values for several tests, the values are, 0.001, 0.01 and 0.1.

Next, for generating the parent outputs, a single vector with all the input points is generated, for the case a vector with fifty values between 0 and 2π. Then, the multiple output covariance matrix for the parent process is computed over this single vector leading to a covariance such as the one in figure 4a. Thus, a sample from a parent GP is generated (figure 4b), the GP has zero mean and covariance $\mathbf{K_g}$ as the generative model in equation (16) states.

Once the parent is generated, the children are sampled using the parent signal as the mean. The covariance matrix is computed using the same input vector for the parent covariance matrix. The kernel $\mathbf{K_f}$ is the one used for this part. Thus, in figure 5b four samples from the children GP are shown. These are the examples of observations taken as inputs for the proposed hierarchical model and the model to be compared with, in this case, a simple Multiple Output GP (MOA) with no hierarchy structure included. The hierarchical model was constructed using an LMC with same amount of ICM as the generated data for each layer, for the case of the MOA model also an LMC was used to construct it.

Results After the data is properly organized the interpolation process initiates defining a number of validation stages. In each validation stage and for each child a random permutation of all the input points indexes is made in order to choose the missing input points in which both models are going to predict. Both models are trained in each validation stage. For the case only one of the children is used as test subject of the reconstruction. Over the test subjects and for all the tests only five of the fifty input points were assumed to be known. All the other children are assumed to be observed data.



(a) Example of covariance for the parent Process. (b) Example of a sample from the parent process.

Figure 4: Covariance/sample of the parent process

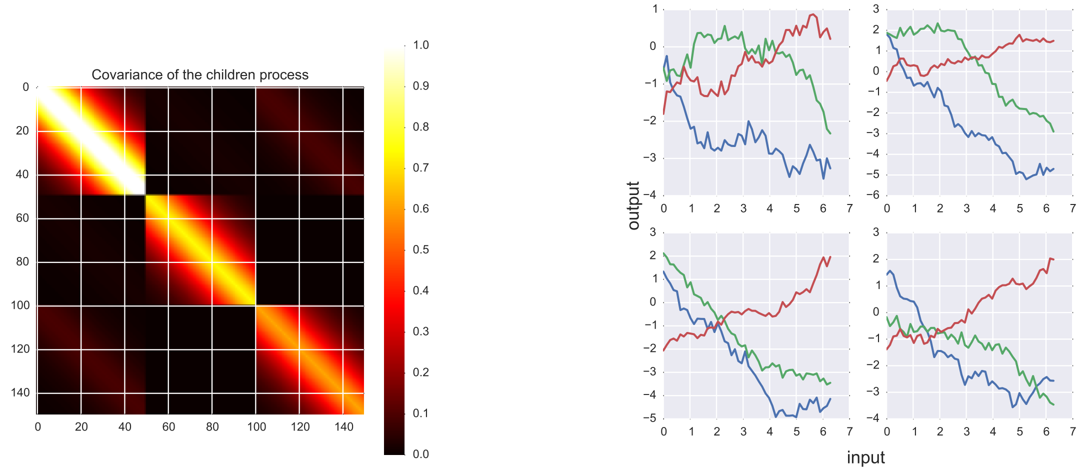
In each validation the Mean Squared Error(MSE), Standarized Mean Squared Error (SMSE) and the Mean Standarized Log Loss (MSLL) are computed. This is the way to measure the prediction accuracy of the MOHGP and MOA. In tables 1, 2 and 3 the results are shown in average over all the validation stages (five for this case). Also, two standard deviation values are computed across validations for each test. Several tests were made changing the amount of children, outputs and noise levels. These parameters were chosen to check the performance of each model within different scenarios. For example, in the presence of noisy signals, when the amount of children was high or if different amount of outputs were present. The parameters were chosen also thinking in possible real noise levels and combinations of output children amounts for the models to be trained in reasonable time. The bold letter shows the average over all tests. The tables show that the Hierarchical model performs better than MOA for interpolation.

Note: The Mean Standarized Log Loss is a form to measure the performance of a model when it gives a predictive distribution, which is the case of the GP models. This measure been formally defined in literature. [2] The model is determined to be better if the value of the MSLL is negative, and close to zero or even positive for models with less quality. The definition Standarized Log Loss in this work is given by:

$$SLL = \frac{1}{2}\left[\log(2\pi{\sigma_*}^2) + \frac{(y_* - \bar{f}(t_*))^2}{{\sigma_*}^2} - \log(2\pi\,\mathrm{var}(\mathbf{y})) - \frac{(y_* - \bar{\mathbf{y}})}{\mathrm{var}(\mathbf{y})}\right], \tag{26}$$

where y_* is the real value, $\bar{f}(t_*)$ is the mean of the predictive distribution at time t_*, ${\sigma_*}^2$ is the predictive variance and $\mathbf{y}$ is the set of observed values.

[2] C. E. Rasmussen and C. K. I. Williams, Gaussian Process for Machine Learning, p. 23. The MIT press, 2006.



(a) Example of covariance for the children process.

(b) Example of four samples from the children process.

Figure 5: Covariance/sample from the children process

Table 1: MSE of interpolation by the models under different noise levels plus/minus two standard deviations. For each test the number of observed points were five out of fifty, chosen randomly for each validation stage. Noise var: variance of white noise, n out: number of outputs of each child, n. child: number of children. MOHGP and MOA constructed with LMC as the sum of three ICM. In bold letter are highlighted the best results for both models and the average across tests.

test	noise var	n out	N. child	MSE MOHGP	MSE MOA
test 1	0.001	3	5	0.097 ± 0.077	1.154 ± 0.033
test 2	0.001	6	6	0.162 ± 0.248	0.710 ± 0.072
test 3	0.001	3	10	**0.066 ± 0.044**	**0.702 ± 0.041**
test 4	0.01	3	5	0.131 ± 0.103	1.181 ± 0.046
test 5	0.01	6	6	0.204 ± 0.233	1.948 ± 0.078
test 6	0.01	3	10	0.107 ± 0.067	0.723 ± 0.074
test 7	0.1	3	5	0.131 ± 0.103	1.195 ± 0.049
test 8	0.1	6	6	0.618 ± 0.147	2.181 ± 0.050
test 9	0.1	3	10	0.599 ± 0.221	1.017 ± 0.055
mean ± std				**0.235 ± 0.406**	**1.201 ± 1.005**

Table 2: SMSE of interpolation by the models under different noise levels plus/minus two standard deviations. For each test the number of observed points were five out of fifty, chosen randomly for each validation stage. Noise var: variance of white noise, n out: number of outputs of each child, n. child: number of children. MOHGP and MOA constructed with LMC as the sum of three ICM. In bold letter are highlighted the best results for both models and the average across tests.

test	noise var	n out	N. child	SMSE MOHGP	SMSE MOA
test 1	0.001	3	5	0.067 ± 0.055	0.789 ± 0.042
test 2	0.001	6	6	0.087 ± 0.137	1.024 ± 0.019
test 3	0.001	3	10	**0.045 ± 0.032**	0.485 ± 0.067
test 4	0.01	3	5	0.091 ± 0.078	0.809 ± 0.078
test 5	0.01	6	6	0.108 ± 0.125	1.023 ± 0.021
test 6	0.01	3	10	0.074 ± 0.052	0.496 ± 0.082
test 7	0.1	3	5	0.091 ± 0.078	0.819 ± 0.081
test 8	0.1	6	6	0.289 ± 0.075	1.020 ± 0.012
test 9	0.1	3	10	0.135 ± 0.051	**0.228 ± 0.013**
mean ± std				**0.109 ± 0.135**	**0.743 ± 0.532**

Table 3: MSLL of interpolation by the models under different noise levels plus/minus two standard deviations. For each test the number of observed points were five out of fifty, chosen randomly for each validation stage. Noise var: variance of white noise, n out: number of outputs of each child, n. child: number of children. MOHGP and MOA constructed with LMC as the sum of three ICM. In bold letter are highlighted the best results for both models and the average across tests.

test	noise var	n out	N. child	MSLL MOHGP	MSLL MOA
test 1	0.001	3	5	-1.739 ± 0.350	0.927 ± 1.549
test 2	0.001	6	6	**-1.917 ± 0.322**	0.285 ± 0.050
test 3	0.001	3	10	-1.783 ± 0.244	**-0.504 ± 0.069**
test 4	0.01	3	5	-1.444 ± 0.225	0.568 ± 1.206
test 5	0.01	6	6	-1.435 ± 0.251	0.274 ± 0.057
test 6	0.01	3	10	-1.457 ± 0.116	-0.184 ± 1.254
test 7	0.1	3	5	-1.444 ± 0.225	0.400 ± 0.716
test 8	0.1	6	6	-0.706 ± 0.074	0.168 ± 0.023
test 9	0.1	3	10	-1.025 ± 0.245	0.009 ± 1.661
mean ± std				**-1.438 ± 0.713**	**0.216 ± 0.784**

The tests with artificial data made so far were made using both models, MOHGP and MOA, with a similar configuration. Thus, the proposed model was constructed using a LMC as the sum of several ICM, in this case with three ICM for the two layers. The same structure for the kernel was used for the MOA model kernel. However, an additional experiment was made in order to test the performance of the hierarchical model in disadvantage against the MOA model. So, in table 4, 5 and 6 the interpolation performance measures are shown in which the MOHGP model kernel is constructed using only one ICM while the MOA model is left with the sum of three ICM as before. Only five points are observed for the interpolation. Besides, the training of the MOA model is restarted 5 times. This is done to find a better combination of the hyper-parameters for the MOA model only. The artificial data was generated by the same method explained in the first part of this section.

Table 4: MSE of interpolation of both models. Noise var: variance of white noise. Miss data: all child: the missing data is in all children, one child: the missing data is on one of the children. MOHGP model constructed with ICM while MOA model constructed with LMC and restarting optimization 5 times. In bold letter are highlighted the best results for both models and the average across tests.

test	noise var	Miss data	MSE MOHGP	MSE MOA
test 1	0.001	All child	**0.162 ± 0.101**	**0.983 ± 0.190**
test 2	0.01	All child	0.229 ± 0.142	1.061 ± 0.164
test 3	0.1	All child	0.621 ± 0.215	1.290 ± 0.275
test 4	0.001	One child	0.240 ± 0.507	1.184 ± 0.052
test 5	0.01	One child	0.296 ± 0.520	1.216 ± 0.062
test 6	0.1	One child	0.630 ± 0.187	1.511 ± 0.064
mean ± std			**0.363 ± 0.379**	**1.207 ± 0.338**

Table 5: SMSE of interpolation of both models. Noise var: variance of white noise. Miss data: all child: the missing data is in all children, one child: the missing data is on one of the children. MOHGP model constructed with ICM while MOA model constructed with LMC and restarting optimization 5 times. In bold letter are highlighted the best results for both models and the average across tests.

test	noise var	Miss data	SMSE MOHGP	SMSE MOA
test 1	0.001	all child	**0.045 ± 0.028**	**0.271 ± 0.055**
test 2	0.01	all child	0.063 ± 0.038	0.291 ± 0.046
test 3	0.1	all child	0.138 ± 0.048	0.286 ± 0.061
test 4	0.001	One child	0.168 ± 0.358	0.819 ± 0.020
test 5	0.01	One child	0.206 ± 0.367	0.842 ± 0.031
test 6	0.1	One child	0.141 ± 0.045	0.338 ± 0.027
mean ± std			**0.126 ± 0.112**	**0.474 ± 0.505**

Table 6: MSLL of interpolation of both models. Noise var: variance of white noise. Miss data: all child: the missing data is in all children, one child: the missing data is on one of the children. MOHGP model constructed with ICM while MOA model constructed with LMC and restarting optimization 5 times. In bold letter are highlighted the best results for both models and the average across tests.

test	noise var	Miss data	MSLL MOHGP	MSLL MOA
test 1	0.001	all child	**-1.961 ± 0.084**	-0.427 ± 0.683
test 2	0.01	all child	-1.445 ± 0.540	**-0.448 ± 0.278**
test 3	0.1	all child	-0.484 ± 2.266	-0.407 ± 0.336
test 4	0.001	One child	-1.596 ± 0.474	1.007 ± 1.588
test 5	0.01	One child	-1.319 ± 0.507	-0.324 ± 0.017
test 6	0.1	One child	-0.963 ± 0.245	1.152 ± 1.777
mean ± std			**-1.294 ± 0.939**	**0.092 ± 1.400**

According to tables 4, 5 and 6 the hierarchical model keeps being superior, despite the fact the MOA model was trained with several advantages. these tests suggest also that the hierarchical model is better when data is presumed to have a parent children structure. To go beyond with testing the performance of the MOHGP model, some other experiments were made. If one checks the results of tables referenced before, it is interesting to see that when the noise is higher the less the difference between the performance of MOHGP and MOA. Thus, a final test was made using and ICM to generate the data but using the same process as before. Then, a white noise was added with 0.1 variance. For this case the amount of observed points were increased to ten and twenty. This is done to determine whether increasing the amount of observed points improves the accuracy of MOA. The same advantages given before to MOA model were given also in this test, except that for this time the optimization process was restarted 10 times for MOA to find a better combination of the hyper-parameters. In tables 7, 8 and 9 the results for this setting are condensed. Is interesting to see that in table 9 the MSLL is better for the MOA model, however for the MSE and SMSE the hierarchical model continues as superior, a further discussion about these results is made in section 7.

Table 7: MSE of interpolation of both models. Noise var: variance of white noise. Miss data: all child: the missing data is in all children, one child: the missing data is on one of the children. MOHGP model constructed with ICM while MOA model constructed with LMC and restarting optimization 10 times. In bold letter are highlighted the best results for both models and the average across tests.

test	N obs.	Miss data	MSE MOHGP	MSE MOA
test 1	10	All child	0.095 ± 0.008	0.142 ± 0.008
test 2	10	One child	0.106 ± 0.011	0.168 ± 0.020
test 3	20	All child	**0.078 ± 0.005**	**0.136 ± 0.007**
test 4	20	One child	0.100 ± 0.025	0.173 ± 0.036
mean ± std			**0.094 ± 0.020**	**0.154 ± 0.031**

Table 8: SMSE of interpolation of both models. Noise var: variance of white noise. Miss data: all child: the missing data is in all children, one child: the missing data is on one of the children. MOHGP model constructed with ICM while MOA model constructed with LMC and restarting optimization 10 times. In bold letter are highlighted the best results for both models and the average across tests.

test	N obs.	Miss data	SMSE MOHGP	SMSE MOA
test 1	10	all child	0.115 ± 0.008	0.172 ± 0.012
test 2	10	One child	0.105 ± 0.013	0.168 ± 0.021
test 3	20	All child	0.097 ± 0.009	0.169 ± 0.014
test 4	20	One child	**0.093 ± 0.015**	**0.161 ± 0.032**
mean ± std			**0.102 ± 0.016**	**0.167 ± 0.008**

Table 9: MSLL of interpolation of both models. Noise var: variance of white noise. Miss data: all child: the missing data is in all children, one child: the missing data is on one of the children. MOHGP model constructed with ICM while MOA model constructed with LMC and restarting optimization 10 times. In bold letter are highlighted the best results for both models and the average across tests.

test	N obs.	Miss data	MSLL MOHGP	MSLL MOA
test 1	10	all child	-0.342 ± 1.656	-0.911 ± 0.039
test 2	10	One child	**-0.854 ± 0.178**	-0.942 ± 0.092
test 3	20	All child	-0.490 ± 2.029	-0.927 ± 0.039
test 4	20	One child	-0.771 ± 0.508	**-0.971 ± 0.156**
mean ± std			**-0.614 ± 0.414**	**-0.937 ± 0.044**

5.1.2 MOCAP Data

The data from the CMU MOCAP database[3] consists on a set of signals, each one of them representing a degree of freedom (DoF) of one of the joints from the skeleton of the MOCAP subject. Thus, the skeleton is formed by sixty two outputs grouped in twenty nine joints. Assuming that for each subject N frames are taken, the size of the kernel matrix will be proportional to $(DoF \times N \times subjects)^2$. Thus, each time we add a subject to the hierarchy this matrix grows to a size which can be heavy in terms of memory. It was decided then to use only a two layer hierarchy, and for the case of interpolation only three subjects were processed and several DoF from some joints considered noisy were removed using the signal to noise ratio (SNR) as criterion. Thus, if the log of SNR was negative the signal was considered noisy. Other joints were manually rejected for interpolation such as Clavicle, Fingers, Wrists, Hand center, Thumbs and Toes.

By manually rejecting some joints and removing some of the noisy DoF the amount of outputs was reduced from 62 to 39. Besides the constant signals were eliminated, given that these signals were not giving any information. For the interpolation tests one type of motion was selected, for the case, walking. For this kind of motion, subject 2 and 7 were chosen to use the motions 1,2 and 2 respectively. Subsequently, the pair subject and motion will be referred here by the notation $X(Y)$, where X refers to the subject and Y to the particular motion. Also a time for subsampling was set in order to optimize the memory used. The minimum amount of total subsamples achieved was 90 in order to visualize appropriate motion.[4] The data is normalized before the training process using min max normalization.

Kernel selection Given that the objective of this work is test the performance of the hierarchical model rather than looking for a good kernel for extracting correlations, a Matern32 kernel was used. For the case of the hierarchical model this kernel was used both in the first and second layers. Additionally a white noise was added to the Matern32 kernel used in the second layer. For the MOA model the sum of the Matern32 and white noise kernel was also used.

Cross Validation A cross validation was made for measuring the prediction accuracy of the hierarchical model(MOHGP) and the simple multiple output model(MOA). The interpolation was made over one of the subjects, conditioning to the information given by the other two. Thus, for all the joints of that subject several input time points were taken as observed and the remaining ones used to predict. Five different sets of Observed/predict time points were chosen on each output for that subject. The mean and standard deviation of the prediction for several amounts of observed/predict points were condensed in tables 10 and 11.

[3] The CMU Graphics Lab Motion Capture Database was created with funding from NSF EIA-0196217 and is available at http://mocap.cs.cmu.edu

[4] Frame intervals where chosen for the walking motion. For 2(1), 2(2), 8(7) and 7(2) frames $[0, 298]$. For 10(5), 10(6) and 11(1) frames $[0, 436]$, $[130, 566]$ and $[130, 566]$ respectively

Table 10: Walking: Accuracy performance of interpolation over 2(1) MOHGP(Matern 32 + White Noise) plus/minus 2 standard deviations. N obs: Number of observed time points of subject 2(1). In bold letter is highlighted the average across tests plus/minus 2 standard deviations.

N obs	MSE	SMSE	MSLL
40	0.004 ± 0.001	0.050 ± 0.015	-2.005 ± 0.156
30	0.006 ± 0.002	0.078 ± 0.032	-1.768 ± 0.177
20	0.013 ± 0.002	0.156 ± 0.029	-1.409 ± 0.212
10	0.032 ± 0.011	0.384 ± 0.135	-0.723 ± 0.196
mean ± std	**0.0137 ± 0.022**	**0.167 ± 0.262**	**-1.476 ± 0.967**

Table 11: Walking: Accuracy performance of interpolation over 2(1) MOA (Walking): (Matern 32 + White Noise) plus/minus 2 standard deviations. N obs: Number of observed time points of subject 2(1). In bold letter is highlighted the average across tests plus/minus 2 standard deviations.

N obs	MSE	SMSE	MSLL
40	0.064 ± 0.002	0.771 ± 0.028	0.164 ± 0.033
30	0.070 ± 0.004	0.841 ± 0.034	0.222 ± 0.036
20	0.075 ± 0.006	0.899 ± 0.047	0.269 ± 0.053
10	0.080 ± 0.003	0.967 ± 0.022	0.338 ± 0.027
mean ± std	**0.072 ± 0.011**	**0.869 ± 0.144**	**0.248 ± 0.127**

The results showed that the MOHGP model is better for interpolation of the walking motion given that outperforms the MOA model in each one of the tests. No additional tests were added given that all the models trained in the cross validation took several days to train. However, further tests are suggested to use different kind of motions.

5.2 Synthesis

In this section the results for the new synthesized artificial/MOCAP samples are showed. Initially is required to make some remarks about the synthesis process given that are several ways to sample from the proposed model (MOHGP) and from the model to be compared with (MOA). The same process used in 5.1.1 is used here to generate the observations for training. The performance measures used here are the ones defined in section 4.3 applied for both artificial and MOCAP data.

Synthesis Process MOHGP The process for synthesizing new samples from a trained MOHGP model is based in the generative model in equation (16), however for this case the function $\mathbf{g_n}$ is not generated from the parent process directly but rather is first inferred from the observed data and then is used as mean of the children process to generate the samples.

Synthesis Process MOA For multiple output model there is no hierarchical structure as in the proposed model, however one can see that the covariance matrix computed with this model is analogous to the covariance matrix of the children process of the proposed model which uses as mean the underlying trend of the data. Thus, to compute the underlying trend for the case of MOA there are two possible ways, one is to average each output across all the observed data, and the other one is to use the GP to predict all the outputs given the observations. In this case the second approach was used.

Note: The similarity measures presented in this work in order to establish the quality of the synthesis were proposed by the authors as a quick way to check if the samples generated by some random process were similar to the real samples different from the training set but from the same phenomena. Thus, some clarifications about the results obtained in these sections are made in the appendix A.4. It is recommended for the reader to review these clarifications before going forward. The reader should be aware that the main objective of this work is not to establish effective correlation measures between motions.

5.2.1 Artificial Data

The data for training was properly generated as explained before in section 5.1.1, nevertheless the coregionalization matrices were slightly modified to add more correlation on the outputs and the amount of inputs was reduced to thirty points. After this, both models were trained and used to generate samples using the process explained in the beginning of this section. As well as in the interpolation section the tests were made varying almost the same parameters, noise variance, amount of outputs and number of observations. In tables 12 and 13 the performance measures defined in this section are shown for both models. The control model is made in order to assure that the correlations measures makes sense, see section A.4.1.

Table 12: MCC of synthesis by the models under different noise levels. noise var: variance of white noise, n out: number of outputs of the observed children, N. child: number of observed children. In bold letter are highlighted the best results for both models and the average across tests.

test	noise var	n out	N. child	MCC MOHGP	MCC MOA	Control
test 1	0.001	3	5	10.571 ± 0.875	**9.931 ± 0.826**	7.119 ± 0.728
test 2	0.001	6	6	9.892 ± 0.797	9.264 ± 0.753	7.522 ± 0.692
test 3	0.001	3	10	**10.719 ± 0.898**	9.481 ± 0.772	7.119 ± 0.728
test 4	0.01	3	5	10.569 ± 0.882	9.578 ± 0.795	7.144 ± 0.811
test 5	0.01	6	6	9.971 ± 0.821	9.188 ± 0.745	7.494 ± 0.704
test 6	0.01	3	10	10.685 ± 0.932	9.850 ± 0.590	7.183 ± 0.827
test 7	0.1	3	5	9.952 ± 1.003	9.554 ± 0.800	7.136 ± 0.945
test 8	0.1	6	6	9.627 ± 0.782	8.981 ± 0.750	7.494 ± 0.704
test 9	0.1	3	10	9.994 ± 1.015	9.363 ± 0.965	7.246 ± 0.938
mean ± std				**10.220 ± 0.775**	**9.465 ± 0.576**	**7.273 ± 0.334**

Table 13: MLC of synthesis by the models under different noise levels. noise var: variance of white noise, n out: number of outputs of the observed children, N. child: number of observed children. In bold letter are highlighted the best results for both models and the average across tests.

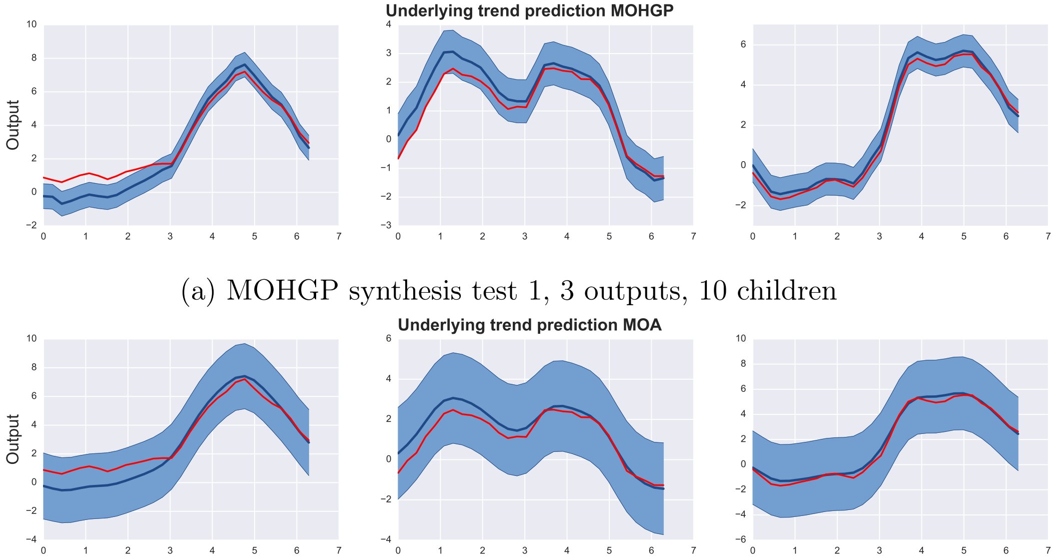
test	noise var	n out	N. child	MLC MOHGP	MLC MOA	control
test 1	0.001	3	5	0.870 ± 0.084	**0.736 ± 0.072**	-0.117 ± 0.089
test 2	0.001	6	6	0.796 ± 0.117	0.625 ± 0.105	-0.069 ± 0.061
test 3	0.001	3	10	**0.886 ± 0.104**	0.652 ± 0.070	-0.117 ± 0.089
test 4	0.01	3	5	0.874 ± 0.086	0.687 ± 0.067	-0.114 ± 0.089
test 5	0.01	6	6	0.809 ± 0.090	0.603 ± 0.075	-0.020 ± 0.038
test 6	0.01	3	10	0.890 ± 0.099	0.674 ± 0.067	-0.114 ± 0.088
test 7	0.1	3	5	0.830 ± 0.087	0.652 ± 0.070	-0.111 ± 0.091
test 8	0.1	6	6	0.763 ± 0.074	0.561 ± 0.055	-0.020 ± 0.038
test 9	0.1	3	10	0.842 ± 0.095	0.655 ± 0.069	-0.067 ± 0.062
mean ± std				**0.846 ± 0.088**	**0.649 ± 0.094**	**-0.083 ± 0.077**

The results condensed in tables 12 and 13 show that the proposed model performs better than the simple multiple output model for generating new samples similar to the artificially generated data. For example the linear correlation between samples generated from the hierarchical model and the real ones are almost one, which is a great evidence that the samples are drawn correctly according to the model used to generate data. However it must be remarked that given the synthesis process described before, involving the prediction of the underlying tendency first, both models were almost equally successful in this matter, see figure 6 illustrating the underlying trend prediction by both models in one of the tests.

5.2.2 MOCAP Data

For real data several MOHGP models and MOA models were properly trained for different kinds of motions. As well as in the interpolation part the walking motion was chosen here. Besides, other motions were chosen, such as, walking exaggerated and soccer shooting. The same process for joint rejection was used. For the Walking exaggerated motion subject 2, 7 and 8 were chosen to use the motions 1,2 and 7 respectively. The latter walks with an exaggerated stride. For the soccer shooting motion were chosen subjects 10 and 11, choosing the motions 5, 6 and 1. As well as in the interpolation 90 subsamples were extracted for each training subject. [5] In a first scenario the training of the models was made similarly as in the interpolation part, taking three subjects and sub sampling them to process only ninety out of the total frames. However a preliminary visual test for walking showed that the sampled motions generated with this scheme were not close to a real walking motion. This preliminary test was made using three subjects walking for training, and a kernel using only an ICM.

[5] For this additional motions the following frames were chosen: For walking exaggerated subjects 2(1), 8(7) and 7(2) frames $[0, 298]$. For soccer shooting 10(5), 10(6) and 11(1) frames $[0, 436]$, $[130, 566]$ and $[130, 566]$ respectively



(a) MOHGP synthesis test 1, 3 outputs, 10 children

(b) MOA Synthesis Test 1, 3 outputs, 10 children

Figure 6: Underlying trend prediction by both models, MOHGP and MOA. Blue: Prediction, Red: Real parent signal

Table 14: Walking: MCC and MLC comparison of synthesis. Two versions of the proposed model. Ver 1: ICM for the multiple output kernel, ver 2: LMC for the multiple output kernel, in this case a sum of 3 ICM. In bold letter are highlighted the best results for both models and the average across tests.

Test subject	MCC ver 1	MCC ver 2	MLC ver 1	MLC ver 2
subject 12(1)	20.740 ± 1.117	21.131 ± 0.853	0.003 ± 0.104	0.055 ± 0.078
subject 16(15)	21.570 ± 1.150	24.710 ± 0.820	0.007 ± 0.076	**0.437 ± 0.048**
subject 38(2)	21.182 ± 1.041	18.914 ± 0.774	0.055 ± 0.072	-0.238 ± 0.064
subject 35(1)	**22.546 ± 1.257**	**25.248 ± 0.898**	0.025 ± 0.091	0.399 ± 0.051
subject 39(4)	21.787 ± 1.352	24.187 ± 0.743	**0.063 ± 0.102**	0.392 ± 0.056
mean ± std	**21.565 ± 1.213**	**22.838 ± 4.854**	**0.031 ± 0.049**	**0.209 ± 0.526**

Table 15: Walking: MCC and MLC comparison of synthesis of both models. In bold letter are highlighted the best results for both models and the average across tests. In red letter are highlighted the worst results of MOHGP against MOA.

Test subject	MCC MOHGP	MCC MOA	MLC MOHGP	MLC MOA
subject 12(1)	21.131 ± 0.853	20.725 ± 2.227	0.055 ± 0.078	0.027 ± 0.211
subject 16(15)	24.710 ± 0.820	22.373 ± 1.620	**0.437 ± 0.048**	**0.163 ± 0.120**
subject 38(2)	18.914 ± 0.774	20.101 ± 1.738	-0.238 ± 0.064	-0.064 ± 0.204
subject 35(1)	**25.248 ± 0.898**	**23.082 ± 1.686**	0.399 ± 0.051	0.147 ± 0.156
subject 39(4)	24.187 ± 0.743	22.073 ± 1.561	0.392 ± 0.056	0.143 ± 0.152
mean ± std	**22.838 ± 4.854**	**21.671 ± 2.192**	**0.209 ± 0.526**	**0.083 ± 0.176**

In Table 14 the synthesis performance measures are made for two hierarchical model versions. The first version of the model was trained with subjects $2(1), 2(2)$ and $7(2)$ and was made as explained before, with an ICM. The second version was made with only two subjects, 2(1) and 2(2), for the training part and a LMC for the output correlation. Here the LMC was constructed as the sum of several ICM. The results show that increasing the amount of ICM kernels helps to improve the synthesis process. In figure 8 the motion sequence for a synthesized subject with the model presented here is shown, for the case the amount of time points remained the same. The outputs signals rejected before the training are properly assembled for visualization purposes. There are implementation constraints in terms of memory that made difficult an implementation using more training subjects. In section 7 this issue is further discussed.

The results in table 14 show that the second version of the model is better in average for all the test subjects. Also, and more importantly according to the criteria in section A.4.2 in the best case is much better than the first version. The test subjects were chosen from the same MOCAP database, for the case of walking they are completely different from the ones used for training.[6] However in subject 38(2) the first version is better, this can be explained because this subject exhibits a walking behaviour in which the hands remained static, and the subjects generated from the first model version have usually a motion with and static-like behaviour, an example of this can be seen in figure 7.

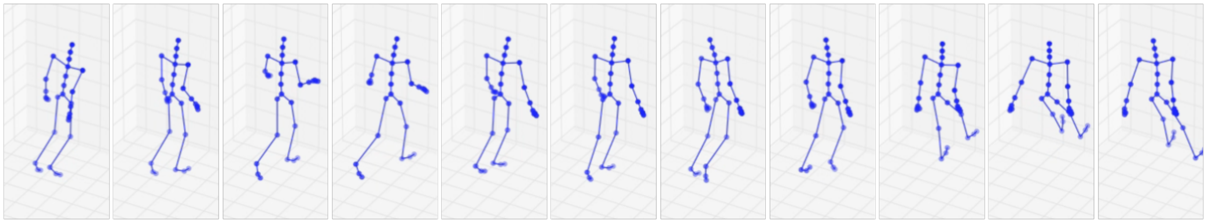


Figure 7: Example of a MOCAP sample from the first hierarchical model version with ICM. the generated subject exhibits an static-like walking behaviour, even if it moves some of the limbs.

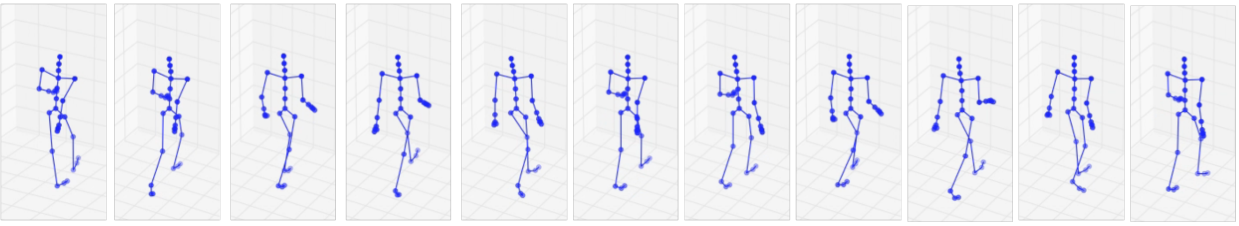


Figure 8: Example of a sample from the proposed model for walking motion.

After realizing that the synthesis process improved using and LMC instead of an ICM, the LMC was used as the appropriate multiple output correlation modelling for the techniques compared here. In table 15 the results for synthesize walking motion are compared between the MOHGP model and the MOA, in this both training subjects are 2(1) and 2(2). The results for walking exaggerated are condensed in table 16. In this case the same subjects used for walking motion were used, given that the only subject(motion) walking with an exaggerated stride was the 8(7). Thus, the training subjects are 2(1) and 8(7). In this case one will expect the generated subjects to be moving between

[6] All these subjects were used in frames $[0, 358]$ with only 90 samples

an exaggerated and a normal way. Finally in table 17 the results for soccer shooting are shown[7], for this case the training subjects are 10(5) and 10(6). It was ensured that the motion in both cases started closely at the same time, the same was ensured for all the other analyzed motions.

Table 16: Walking Exaggerated: MCC and MLC comparison of synthesis for both models. In bold letter are highlighted the best results for both models and the average across tests. In red letter are highlighted the worst results of MOHGP against MOA.

Test subject	MCC MOHGP	MCC MOA	MLC MOHGP	MLC MOA
subject 12(1)	19.943 ± 1.002	20.381 ± 1.737	0.055 ± 0.078	-0.032 ± 0.159
subject 16(15)	24.330 ± 1.180	22.635 ± 1.080	**0.437 ± 0.048**	**0.187 ± 0.102**
subject 38(2)	19.723 ± 1.097	20.400 ± 1.506	-0.238 ± 0.064	-0.048 ± 0.179
subject 35(1)	**25.134 ± 1.311**	**23.414 ± 1.178**	0.399 ± 0.051	0.174 ± 0.128
subject 39(4)	24.090 ± 1.186	22.205 ± 1.097	0.392 ± 0.056	0.161 ± 0.118
mean ± std	**22.644 ± 4.644**	**21.807 ± 2.440**	**0.189 ± 0.519**	**0.088 ± 0.211**

Table 17: Soccer Shooting: MCC and MLC comparison of synthesis for both models. In bold letter are highlighted the best results for both models and the average across tests. In this case, the best are different from the training subjects. In red letter are highlighted the worst results of MOHGP against MOA.

Test subject	MCC MOHGP	MCC MOA	MLC MOHGP	MLC MOA
subject 10(5)	27.068 ± 0.933	25.541 ± 2.812	0.530 ± 0.064	0.378 ± 0.212
subject 10(3)	23.689 ± 1.021	23.377 ± 2.242	-0.035 ± 0.075	0.001 ± 0.111
subject 10(2)	24.776 ± 0.958	23.806 ± 2.726	0.221 ± 0.075	0.165 ± 0.203
subject 10(6)	25.851 ± 0.875	24.500 ± 2.389	0.428 ± 0.065	0.298 ± 0.171
subject 11(1)	**26.085 ± 0.971**	**24.984 ± 2.941**	**0.389 ± 0.067**	**0.297 ± 0.219**
mean ± std	**25.494 ± 2.321**	**24.442 ± 1.560**	**0.307 ± 0.396**	**0.228 ± 0.265**

The overall results shown that the proposed model is much better generating subjects (according to the criteria defined in A.4.2), in figures 9 and 10 two of the subjects generated by the hierarchical model for each one of the remaining motions is shown as an image sequence. Some of the results for the proposed model were highlighted in red. These results corresponds to the low performance test of the MOHGP model against the MOA model. In section 7 these results are further discussed.

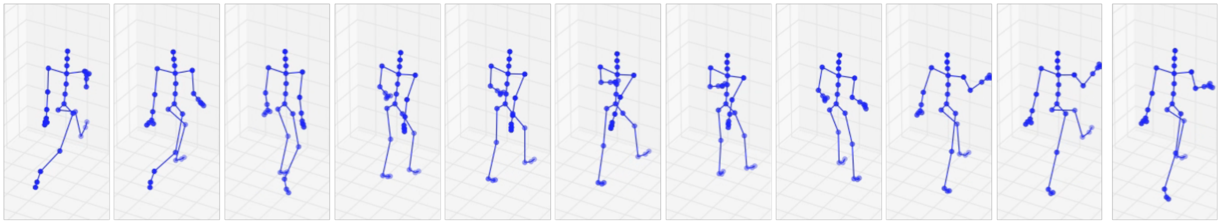


Figure 9: Example of a sample from the proposed model for walking exaggerated motion.

[7] There was not enough subjects of soccer shooting to compare here. 10(5) taken at frames $[0, 232]$, $10(3), 10(2), 10(6)$ and 11(1) taken at frames $[130, 362]$

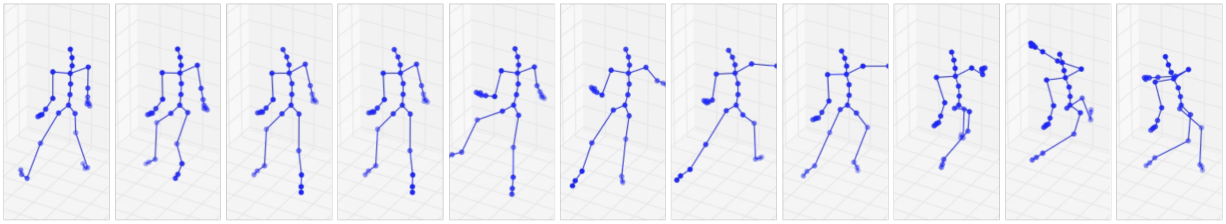


Figure 10: Example of a sample from the proposed model for soccer shooting motion.

5.2.3 Parent and children process

Like in figures 6a and 6b the underlying trend was shown for one of the tests, it is also interesting to see the estimated parent and children process by the hierarchical model extension proposed here. For the case one experiment was performed in order to ilustrate the performance of the proposed model in recovering the process used to generate the artificial data. Thus, the artificial data was generated using the same process as in section 5.1.1. However, the coregionalization matrices were forced to have only random values between 0 and 1 in order to force a bigger correlation between outputs. The white noise added to children kernel was left with value of 0.001.

After generating the data, the MOHGP model and the MOA model were properly trained, each training process was restarted 10 times. in 11a, 11b and 11c the covariance matrices for the children process are shown. One can see that the proposed model recovers a covariance more similar to the real that the one recovered by the MOA model.

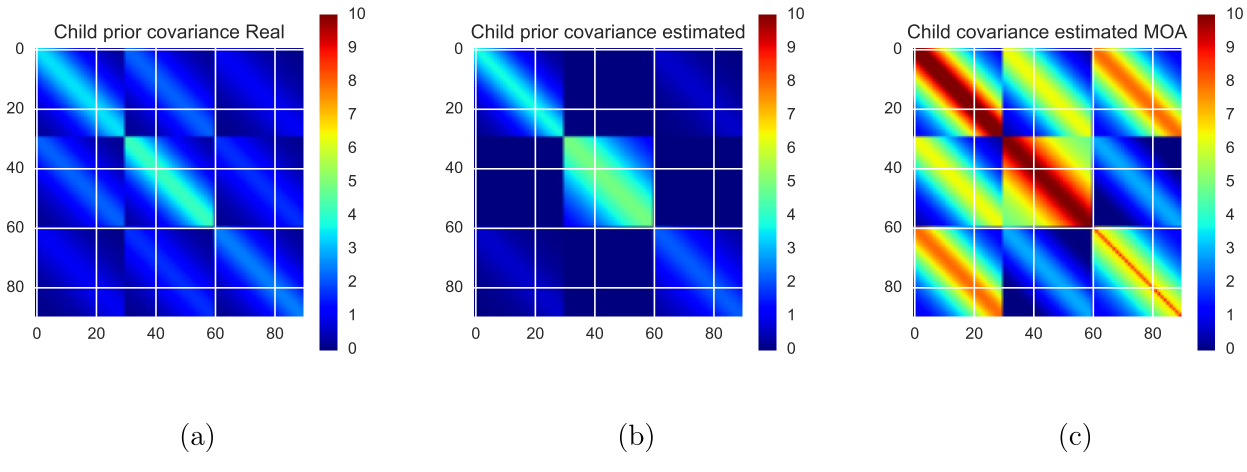
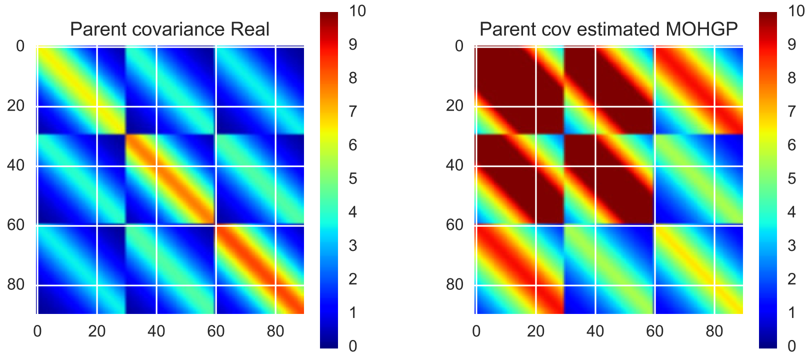


Figure 11: (a) Covariance of the children process to generate the artificial data. (b) Covariance of the children process estimated by the proposed model (MOHGP). (c) Covariance of the children process estimated by the model used to compare (MOA).

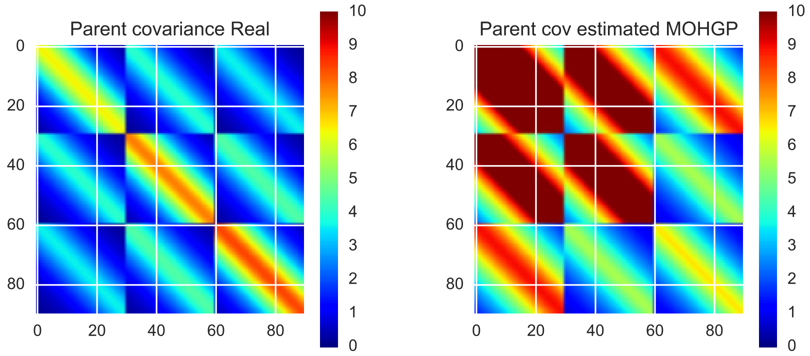
For the parent process, only the covariance estimated by the proposed model is shown. This is due to fact that the MOA model only has one kernel involved. In this case the covariance is directly related to the children, given that initially there is no parent/children structure defined in this model. In



(a) (b)

Figure 12: (a) Covariance of the parent process to generate the artificial data. (b) Covariance of the parent process estimated by the proposed model (MOHGP).

figure 12a and 12b the covariances matrices for the parent process are presented. However, the estimation of the parent covariance made by the proposed model seems poor in relation to the children process. In that sense the training of the MOHGP model was repeated and restarted more times in order to see if better results were obtained. In figure 13a and 13b the real parent covariance and the estimated one after restarting the training process thirty times to keep the best value are shown.



(a) (b)

Figure 13: (a) Covariance of the parent process to generate the artificial data. (b) Covariance of the parent process estimated by the proposed model (MOHGP) after restarting training 30 times to keep the best value.

One can see that there is no much difference between the estimation of the parent process illustrated in figure 12b and the one in figure 13b. Thus, the experiments made in this section for synthesis with artificial data were made training the models with ten times of multiple restarts. In the case of MOCAP data only one cycle of optimization was performed. The reason for this the considerable amount of time taken for training a hierarchical model. However, and how it was reported in the section before, the results were satisfactory with this scheme.

5.2.4 Coregionalization Matrices over MOCAP data

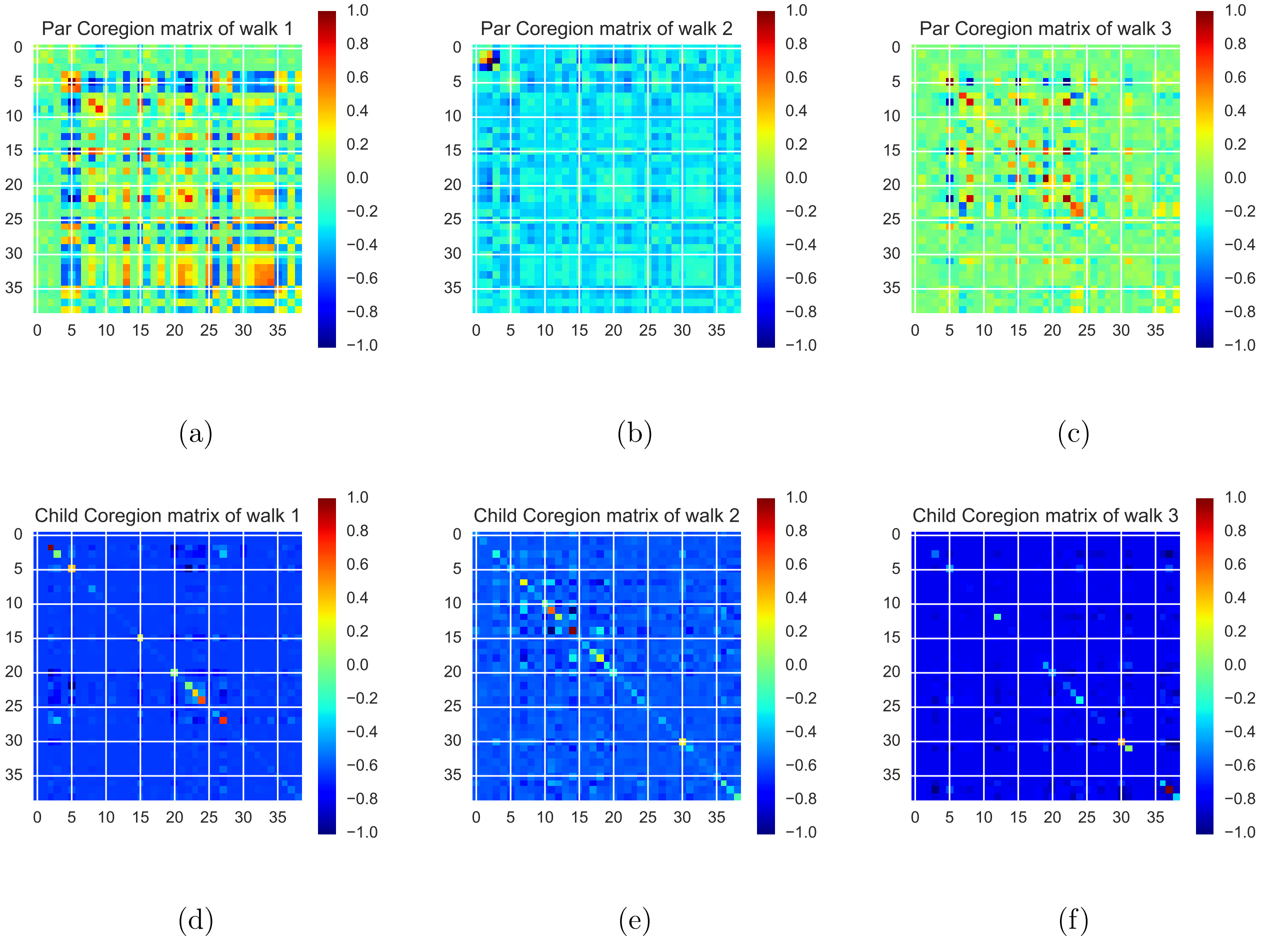
As a matter of illustration and analysis it is important also to show the coregionalization matrices estimated by the training procedure. These matrices encode the correlations between the outputs used to train the model. Thus, these matrices can give some insights about the strong, weak or absence of dependencies between the outputs, in these case the body joints. For the case, the estimated matrices for the walking process are shown. Given that the multiple output model used for training that model was the LMC as the sum of three ICMs, the estimated matrices are six for both parent and child process in the case of the hierarchical model, and three in the case of the MOA model.

In figure 14 all the six coregionalization matrices estimated for both the parent an children process are presented. In this image one can see initially that the correlations captured by the children process are more uniform that in the case of the parent process. One can conclude that all the correlations captured by the children process are negative instead of those in the main diagonal. The parent process captures more positive values, exhibing strong positive correlations between some of the outputs and exhibing more insteresting patterns to analyze.

For example in figure 14b the coregionalization matrix $\mathbf{B}_{g2}$ is shown. while this matrix show weaker positive and negative correlations between almost all of the outputs, exhibits big correlations between the first four outputs which are related with the root joint of the MOCAP skeleton. In this case, the root is located in the center of the body, so one will expect that a change in one of the DoF of this joint will affect other ones as well. However is interesting to see that in the other coregionalization matrices the correlations between those outputs are weaker. This can be explained by the fact that the LMC assumes that different dependencies between the outputs can be happening. Thus, while in one of the coregionalization matrices some joints exhibit a weaker correlation in other ones can be stronger.

Given that a big correlation was observed between the root joints of the body, one will expect also that several joints attached to the same limb to have big correlations. That is the case of the coregionalization matrices in figure 15 in which the correlations between the outputs of the left leg are shown. It is interesting to see that almost all the correlation values imply strong and weak correlations and almost all negative. In this case the left leg joints are two: the left femur (first tree rows), the left tibia (fourth row) and the left foot (fifth row). In this case the mixture of negative correlations and positive correlations can be explained by the fact that when the leg is stretching to walk, the angle of the femur with respect to the root decreases while the angle of the tibia with respect to the knee increases, and this relation changes when the leg retracts itself again. Again, the LMC helps to model different dependencies, that is why probably the synthesis process with ICM was not succesfull as it was presented in figure 7.

Thus, regarding the body limbs with apparently no direct correlation are expected to have weaker correlations. That is the case of the coregionalization matrices represented in figure 16. One can notice that most of the values have low correlations values and in several cases like in figure 16c values equal to zero. However in figure 16f an strong negative correlation is detected.



(a)

(b)

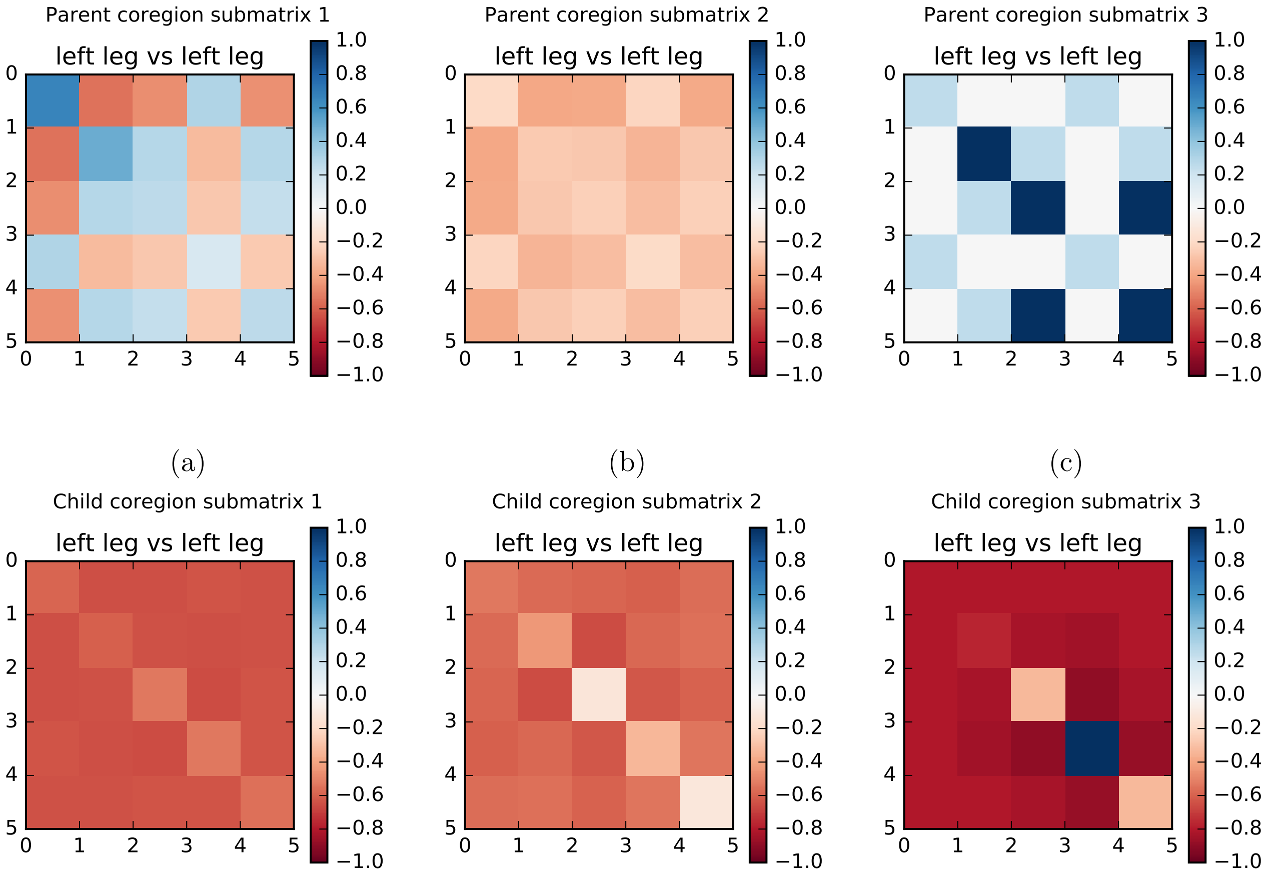
(c)

(d)

(e)

(f)

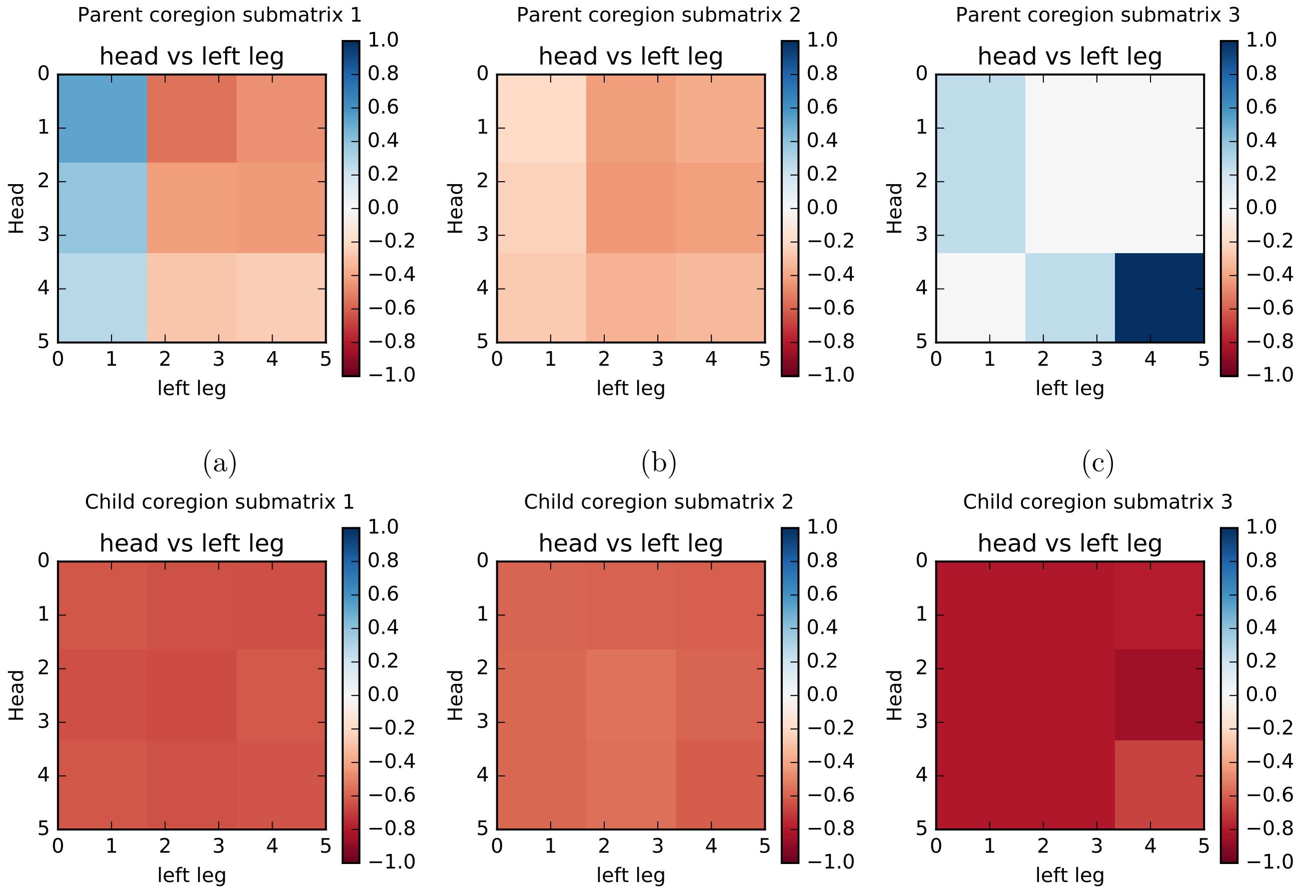
Figure 14: (a)(b)(c) Parent process coregionalization matrices $\mathbf{B}_{g1}, \mathbf{B}_{g2}$ and $\mathbf{B}_{g3}$ estimated from the walking motion data. (d)(e)(f) Child process coregionalization matrices $\mathbf{B}_{f1}, \mathbf{B}_{f2}$ and $\mathbf{B}_{f3}$ estimated from the walking motion data.



(a) (b) (c)

(d) (e) (f)

Figure 15: (a)(b)(c) Parent Coregionalization submatrices, the correlations presented are between the outputs of the left leg. (d)(e)(f) Children Coregionalization submatrices, the correlations presented are between the outputs of the left leg.



(a) (b) (c)

(d) (e) (f)

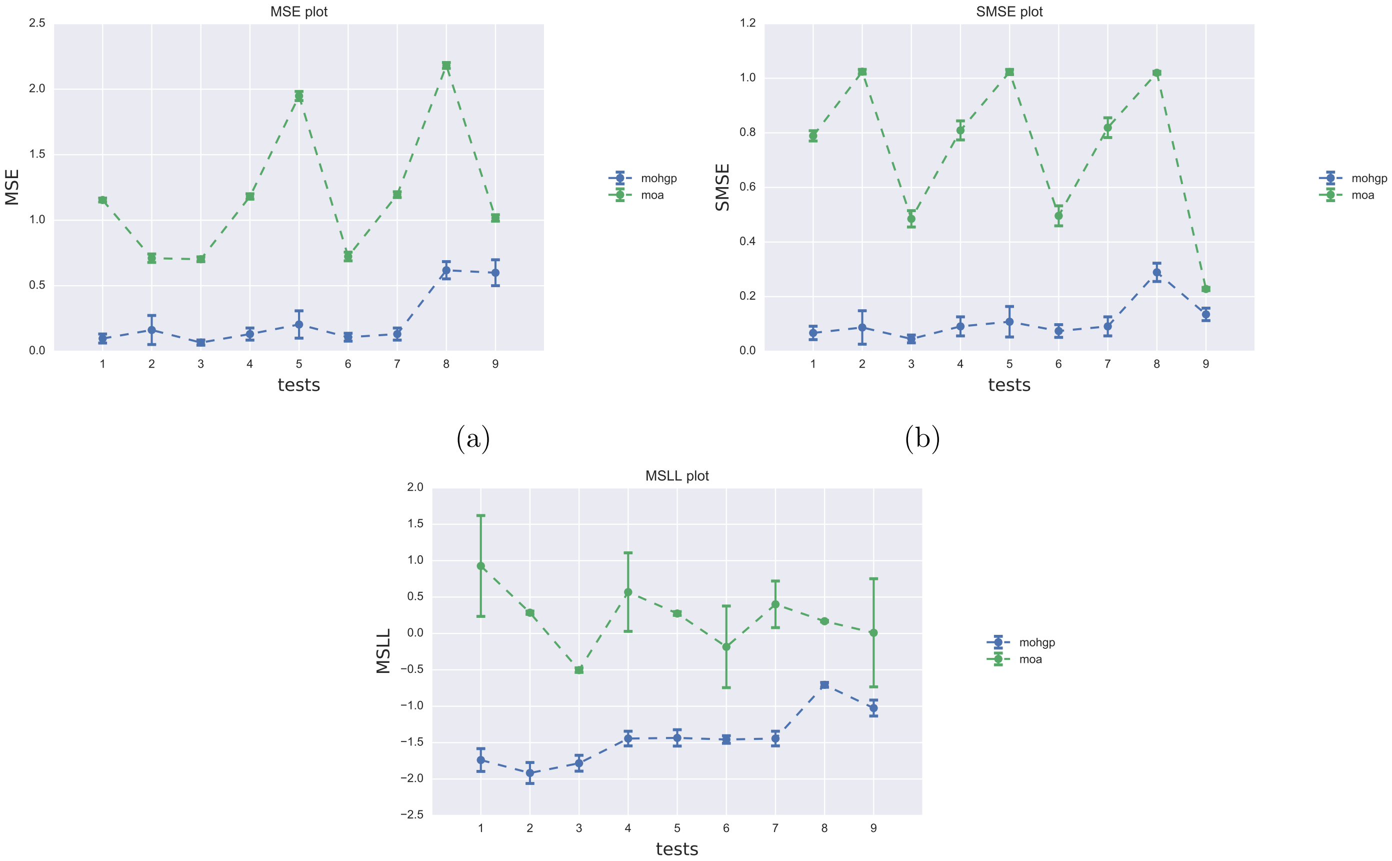
Figure 16: (a)(b)(c) Parent Coregionalization submatrices, the correlations presented are between the outputs of the head and the left femur. (d)(e)(f) Children Coregionalization submatrices, the correlations presented are between the outputs of the head and the left femur.

6 Experimental Results Summary

In this section the error bar plots from each one of the experiments. The reader can verify that the presented model (MOHGP) has a clear trend to overcome the multiple output simple model used here to compare (MOA).

6.0.1 Results over interpolation

The results of the interpolation experiments are shown in this section, each page has three plots. The MSE, SMSE and MSLL with a 95% confidence interval by test.

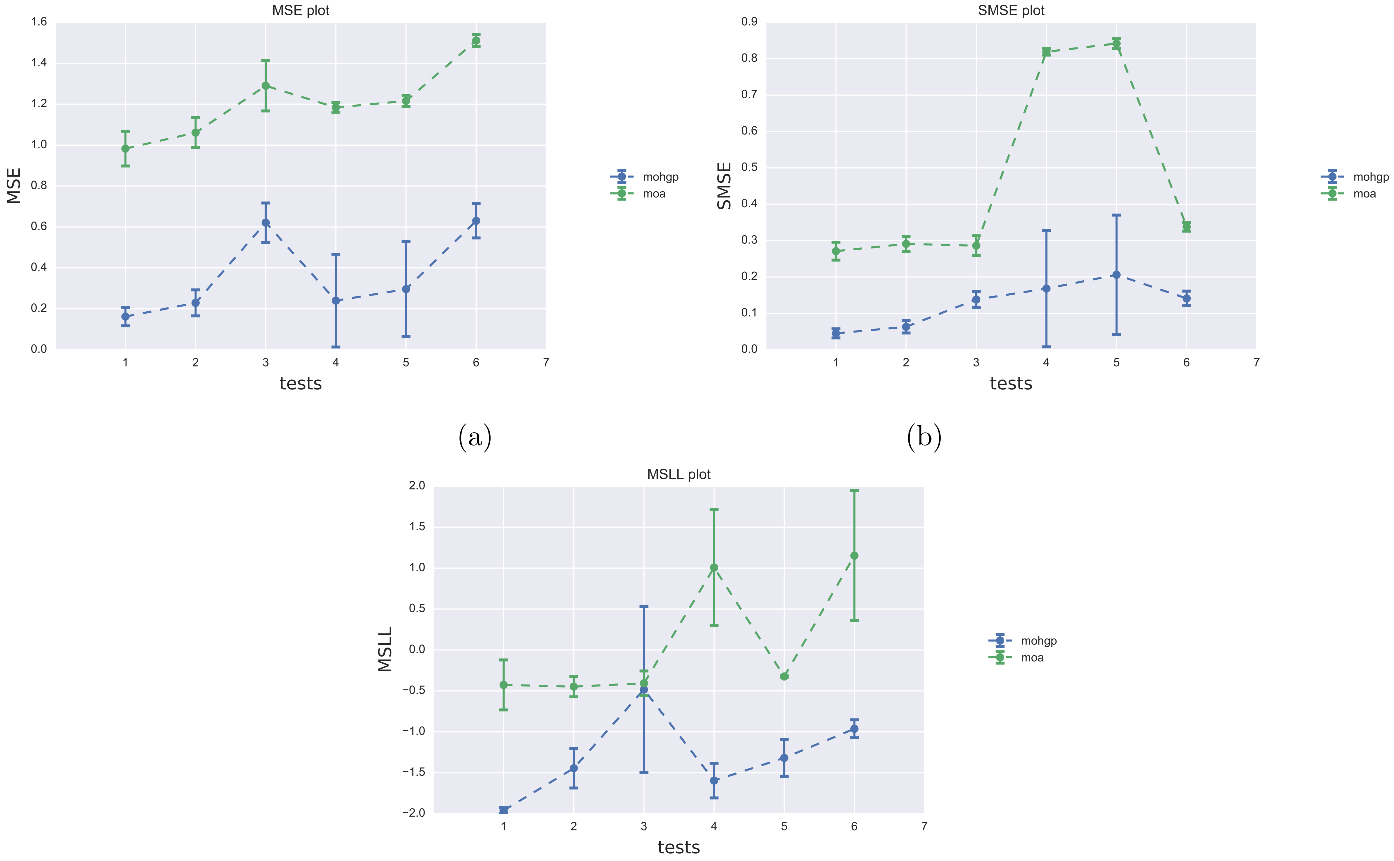


(a)

(b)

(c)

Figure 17: Results of interpolation (from tables 1, 2 and 3): Experiment one on artificial data. (a) MSE of the prediction over one of the subjects with an interval of 95% confidence. (b) SMSE of the prediction over one of the subjects with an interval of 95% confidence. (c) MSLL of the prediction over one of the subjects with an interval of 95% confidence.

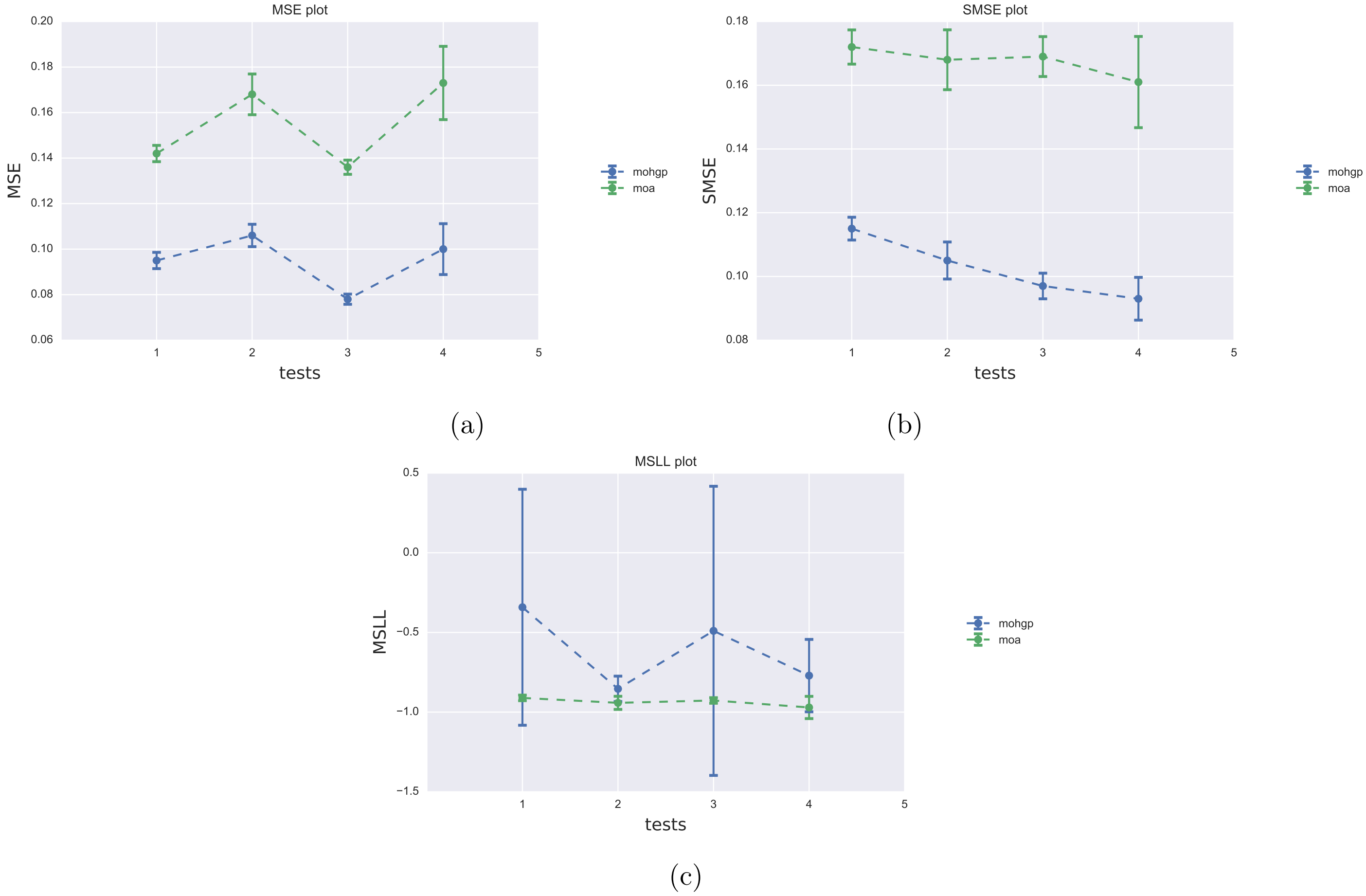


(a)

(b)

(c)

Figure 18: Results of interpolation (from tables 4, 5 and 6): Experiment two on artificial data with MOA tuned. (a) MSE of the prediction over with an interval of 95% confidence. (b) SMSE of the prediction with an interval of 95% confidence. (c) MSLL of the prediction with an interval of 95% confidence.

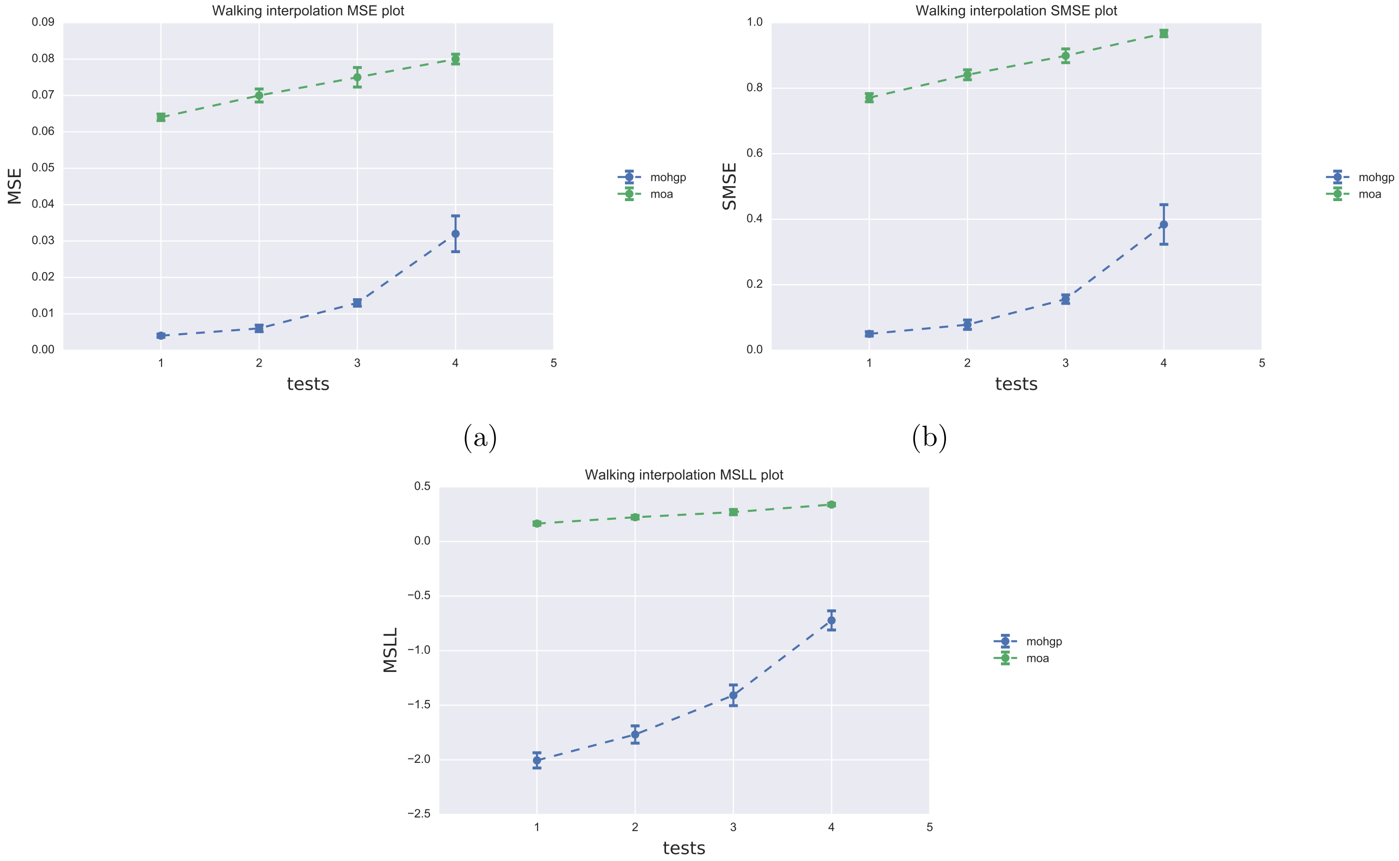


(a)

(b)

(c)

Figure 19: Results of interpolation (from tables 7, 8 and 9): Experiment three on artificial data with MOA tuned. (a) MSE of the prediction with an interval of 95% confidence. (b) SMSE of the prediction with an interval of 95% confidence. (c) MSLL of the prediction with an interval of 95% confidence.



(a)

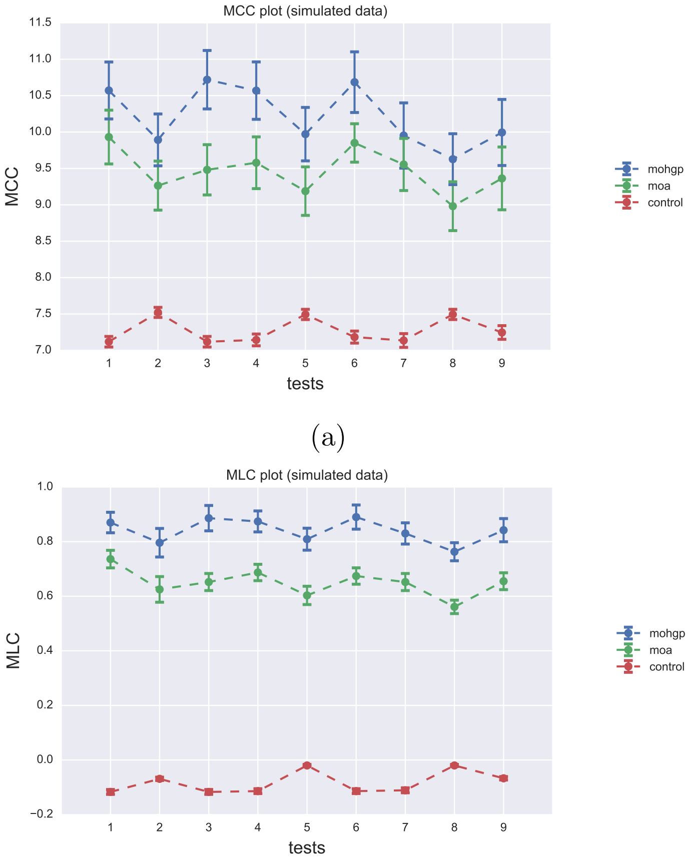
(b)

(c)

Figure 20: Results of interpolation (from tables 10 and 11): Walking. Four tests were performed changing the amount of observed points. (a) MSE of the prediction over one of the walking subjects with an interval of 95% confidence. (b) SMSE of the prediction over one of the walking subjects with an interval of 95% confidence. (c) MSLL of the prediction over one of the walking subjects with an interval of 95% confidence.

6.0.2 Results over Synthesis

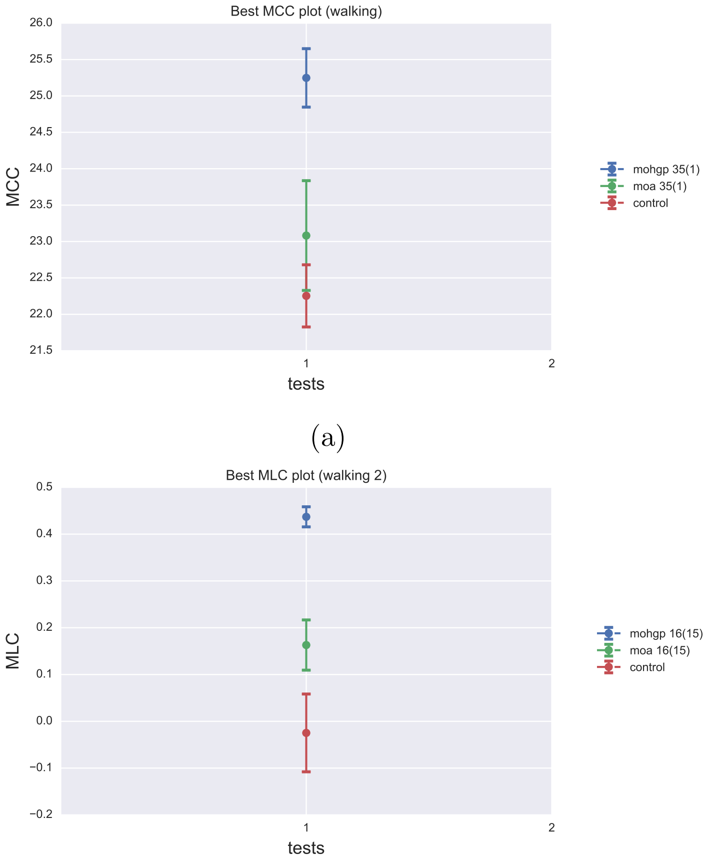
The results of the synthesis experiments are shown in this section. The first two plots correspond to the tests with artificial data, the next six plots correspond to the tests with real data. Here we measure the MCC and MLC of MOHGP and MOA within an interval of 95% confidence for the best matched subject against the control model.



(a)

(b)

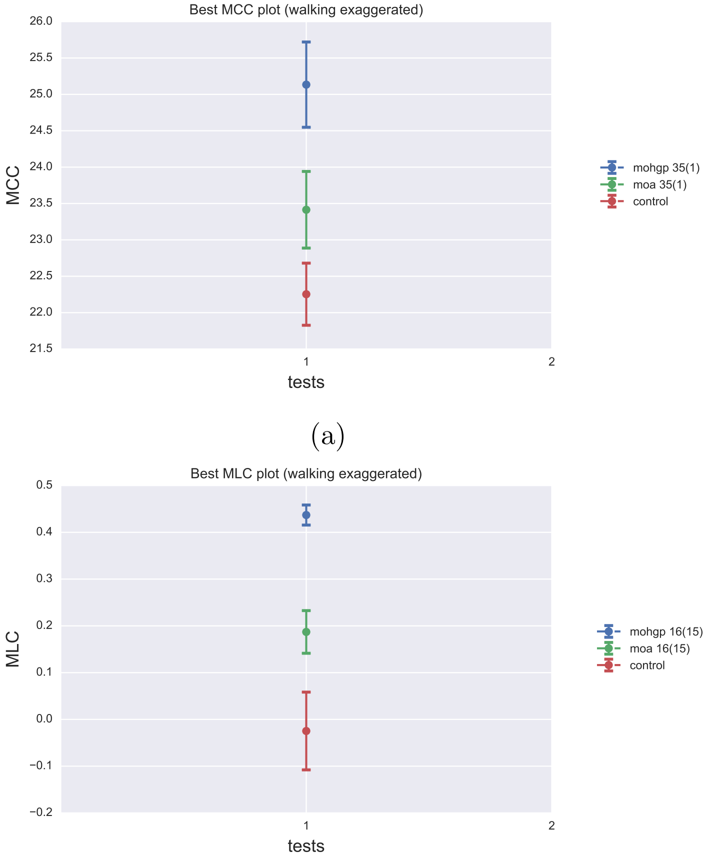
Figure 21: Results of synthesis (from tables 12 and 13): Experiment one with artificial data, varying several parameters to generate children. (a) MCC of the generated samples against the toy model ones in a 95% confidence interval. (b) MLC of the generated samples against the toy model ones in a 95% confidence interval.



(a)

(b)

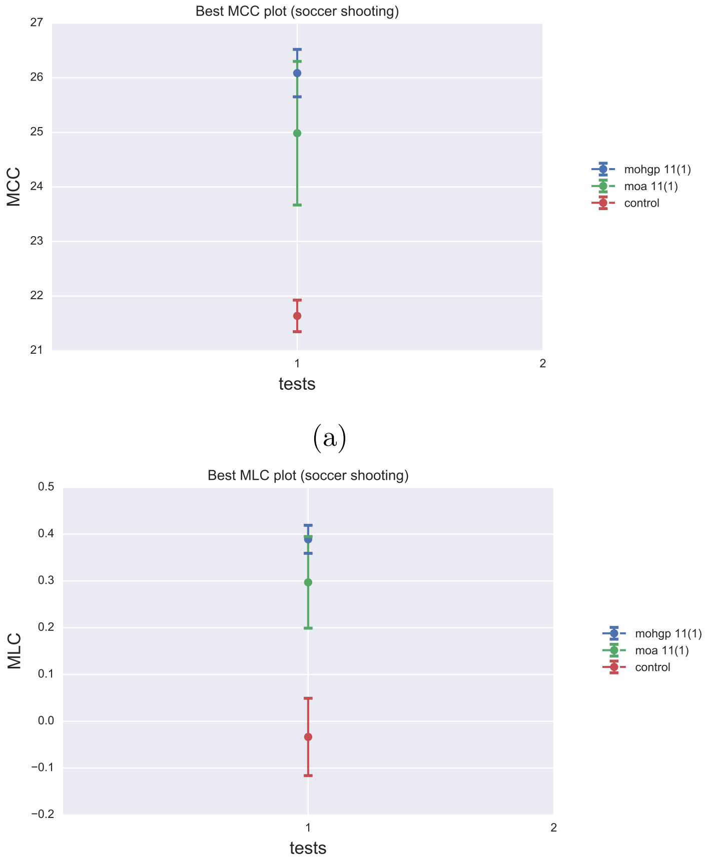
Figure 22: Results of Synthesis (from table 14): Walking. 25 samples were extracted from each model and compared against the real walking subjects. Only the real subjects with the best correlation measure are shown in the plot. (a) MCC of the generated samples against the real ones with an interval of 95% confidence. (b) MLC of the generated samples against the real ones with an interval of 95% confidence.



(a)

(b)

Figure 23: Results of Synthesis (from table 15): Walking Exaggerated. 25 samples were extracted from each model and compared against the real walking subjects. Only the real subjects with the best correlation measure are shown in the plot. (a) MCC of the generated samples against the real ones with an interval of 95% confidence. (b) MLC of the generated samples against the real ones with an interval of 95% confidence.



(a)

(b)

Figure 24: Results of Synthesis (from table 16): Soccer Shooting. 25 samples were extracted from each model and compared against the real soccer shooting subjects. Only the real subjects with the best correlation measure are shown in the plot. (a) MCC of the generated samples against the real ones with an interval of 95% confidence. (b) MLC of the generated samples against the real ones with an interval of 95% confidence.

7 Conclusions and future work

Here a multiple output model was presented as an extension to an existing model for the analysis of observed data to be pressumed to have a replicate structure underneath. Even though the extension of the model involves the addition of Linear Coregionalization Models for the output wise correlation modelling, the extension of the hierarchical model keeps its simplicity, which is demonstrated in appendix A.2. Also the results condensed in section 5 show that the proposed model when is applied to the MOCAP data-set successfully reconstructs missing data (interpolation) given the information of the other subjects used for training, being better than the simple multiple output model (MOA). This is possible thanks to the structure in which the hierarchical model is defined and that it seems to be a valid assumption for motion data.

A parameter estimation procedure was also derived for the proposed model, for the case, the parameters that best fit the model are the ones that maximizes the model likelihood. The derivatives of the model likelihood with respect to each one of the model parameters were generally derived algebraically simplifying the expressions. This lead to a simple computation of the gradient for the model likelihood explained in appendix A.3. Thus, a gradient based optimization algorithm was used in this case to make the optimization of the likelihood function, several toolboxes have this kind of implementations, in this work the GPy library [8] for GPs was used.

One of the more interesting and widely studied problems related with computer-based human animation is the one related with synthesis or generating new motions. First it has to be remarked that the model presented here is a purely data-driven model, no physically inspired prior assumptions were made to train nor the proposed model or the MOA model used to compare. Also, to synthesize new motions, all the models were trained with only two subjects. As showed in the results section using three subjects limited the amount of ICMs used in the parent and children kernels respectively, which limited the behaviour of the signals generated by the model, in some cases generating only static skeletons with no motion involved. Using more ICMs and less subjects was an acceptable way to generate more subjects moving more similar to a human motion.

Several works related with synthesis validate the performance of their synthesis visually rather than in a quantifiable way, despite the fact that a visual evaluation is valuable, even more when an expert is involved, it was valuable also in the setting presented here to count with a quantifiable measure to give an intuition about how much a model was better than other ones. Thus, the MCC and MLC were defined in section 4.3. The results shown that the visual validation and the quantifiable validation here defined are correlated in much of the cases for MOCAP data. in section A.4 some clarifications were made about those correlation measures. The overall conclusion regarding this issue is that more research must be done in order to define a proper correlation measure between the subjects.

The hierarchical model was superior than the MOA model for all the performance measures. However in the interpolation part, two tests were made in which the log loss was better or equal for the MOA model over artificial data. This is explained by the fact that the hierarchical model assumes that the children deviate from the parent by some amount, however those tests were made using a high noise variance for each children and giving advantage to the MOA model in the training part. Also, the over confidence of prediction of the hierarchical model for those tests was penalized by the log loss given the high noisy training examples.

[8] GPy is a Gaussian Process (GP) framework written in python, from the Sheffield machine learning group. Website: http://sheffieldml.github.io/GPy/

For synthesis there are still challenges to generate more realistic motions, some of the generated subjects with the proposed model and specifically in the case of soccer shooting and walking exaggerated motion have some behaviours that do not correspond to an adequate human motion. For example, sometimes the knee bends in an opposite manner to the motion constraints, also there are some generated samples that exhibit a dissonance between the movement of the arms and the movement of the legs. One example of that can be noticed in figure 9, the left leg stretches in an exaggerated manner while the right leg does not, this is related maybe to the fact that one of the training subjects for this generated motion is walking in an exaggerated manner while the other one is walking in a normal way.

Future Works The issues explained in the last paragraph justify the search of several solutions to overcome those challenges that can be tackled in future endeavours. The challenges are listed and explained as follows:

- **Memory constraints**: As explained before, the memory available for the model training process is limited, for this case only three subjects were used as maximum for training with an ICM for the multiple output modelling, also was possible to use only two subjects and increase the amount of ICM to have an LMC for the multiple output modelling, being capable then of synthesizing more realistic motions. However this constraint can be overcome developing a Sparse GP version of the model presented here, which is perfectly possible, and in the literature there are several methods that perfectly fits in this scheme. Another possibility is to try a distributed strategy for GPs.

- **More realistic and stylistic motions**: This issue can be overcome solving the first one given that the more subjects able to be added to the hierarchical model the more variance can be captured and the more accurate the optimization will be to fit the parameters of the model. In addition, this model could be combined with physically inspired prior assumptions through the covariance functions, in [49] there is an example of how to combine physical assumptions with Gaussian Process for human motion data.

- **Deeper hierarchies**: More layers can be added to the model presented here if the first challenge is overcame. Thus, different kinds of motions, like walking and running can be processed in the same hierarchy leading to predict a underlying trend between these different kind of movements and learning a generative model for motion styles combining the features of walking, running, and so on.

- **Other applications**: The model presented here can be applied in other kind of applications also related with computer graphics. For example for human face character generation. One might think in taking several realizations of human facial expressions from the same kind, like laughing, then train a hierarchical model for this data and finally try to generate different kind of laughing expressions given the information learned from the other faces.

A Appendix

A.1 Single Output Hierarchical GP Likelihood Demonstrations

Two points t and t' in the same child signal are jointly Gaussian distributed with zero mean and covariance $k_g(t,t') + k_f(t,t')$, which can be shown in the following way:

According to the generative model in equation (5) one can describe the signal in the time t and in the time t' as :

$$\begin{aligned} f_{nr}(t) &\sim g_n(t) + \mathcal{GP}(0, k_f(t,t)), \\ f_{nr}(t') &\sim g_n(t') + \mathcal{GP}(0, k_f(t',t')). \end{aligned} \tag{27}$$

at the same time $g_n(t) \sim \mathcal{GP}(0, k_g(t,t'))$. Thus the joint distribution of $f_{nr}(t)$ and $f_{nr}(t')$ is given by:

$$p(f_{nr}(t), f_{nr}(t')) = \mathcal{GP}(\mathbf{0}, \mathbf{K}_g) + \mathcal{GP}(\mathbf{0}, \mathbf{K}_f), \tag{28}$$

where $\mathbf{K}_g$ and $\mathbf{K}_f$ is the covariance matrices between the input points t and t'. It is evident that the expected value of the joint distribution above is equal to the zero mean vector. Also is straightforward to see that by the properties of the Gaussian the covariance of the joint distribution is given by $\mathbf{K}_g + \mathbf{K}_f$. Nevertheless one can show this property using the procedure below:

Defining $\mathbf{g}_n \sim \mathcal{GP}(\mathbf{0}, \mathbf{K}_g)$ and $\mathbf{h}_{nr} \sim \mathcal{GP}(\mathbf{0}, \mathbf{K}_f)$, $g_n(t)$ as the first entry of $\mathbf{g}_n$, $g_n(t')$ as the second entry of $\mathbf{g}_n$, $h_{nr}(t)$ as the first entry of $\mathbf{h}_{nr}$ and $h_{nr}(t)$ as the first entry of $\mathbf{h}_{nr}$ the covariance of the joint distribution of $f_{nr}(t)$ and $f_{nr}(t')$ can be written as:

$$\begin{aligned} \mathrm{cov}[f_{nr}(t), f_{nr}(t')] &= \mathrm{cov}[g_n(t) + h_{nr}(t), g_n(t') + h_{nr}(t')] \\ &= \mathrm{cov}[g_n(t), g_n(t') + h_{nr}(t')] + \mathrm{cov}[h_{nr}(t), g_n(t') + h_{nr}(t')] \\ &= \mathrm{cov}[g_n(t'), g_n(t)] + \mathrm{cov}[h_{nr}(t'), g_n(t)] + \mathrm{cov}[g_n(t'), h_{nr}(t)] + \mathrm{cov}[h_{nr}(t'), h_{nr}(t)] \\ &= \mathrm{cov}[g_n(t'), g_n(t)] + \mathrm{cov}[h_{nr}(t'), h_{nr}(t)] \\ &= k_g(t,t') + k_f(t,t') \end{aligned} \tag{29}$$

For the case $h_{nr}(t')$ and $g_n(t)$ are generated independently as well as $h_{nr}(t)$ and $g_n(t')$ so $\mathrm{cov}[h_{nr}(t'), g_n(t)] = \mathrm{cov}[g_n(t'), h_{nr}(t)] = 0$. It is straightforward to see that the mean of the joint distribution of two points from different replicates is equal to zero only changing the left side of equation (28) to $p(f_{nr}(t), f_{nr'}(t'))$, now the points are from different replicates. Furthermore the covariance can be derived using a similar procedure like the one above:

$$\begin{aligned}
\operatorname{cov}[f_{nr}(t), f_{nr'}(t')] &= \operatorname{cov}[g_n(t) + h_{nr}(t), g_n(t') + h_{nr'}(t')] \\
&= \operatorname{cov}[g_n(t), g_n(t') + h_{nr'}(t')] + \operatorname{cov}[h_{nr'}(t), g_n(t') + h_{nr'}(t')] \\
&= \operatorname{cov}[g_n(t'), g_n(t)] + \operatorname{cov}[h_{nr'}(t'), g_n(t)] + \operatorname{cov}[g_n(t'), h_{nr'}(t)] + \operatorname{cov}[h_{nr'}(t'), h_{nr}(t)] \\
&= \operatorname{cov}[g_n(t'), g_n(t)] \\
&= k_g(t, t')
\end{aligned} \tag{30}$$

For the case $h_{nr'}(t')$ and $h_{nr}(t)$ are generated independently, despite of the fact they have the same mean, thus $\operatorname{cov}[h_{nr'}(t'), h_{nr}(t)] = 0$.

A.2 Multiple Output Hierarchical GP Likelihood Demonstrations

For the model proposed in this work two points t and t' in the same child signal $\mathbf{Y}_{nr}$ are jointly Gaussian distributed with zero mean and covariance $\mathbf{K}_g(t,t') + \mathbf{K}_f(t,t')$, which can be shown in the following way:

According to the generative model in equation (16) one can describe the signal in the time t and in the time t' as:

$$\begin{aligned}
\mathbf{f}_{nr}(t) &\sim g_n(t) + \mathcal{GP}(\mathbf{0}, \mathbf{K}_f(t,t')), \\
\mathbf{f}_{nr}(t') &\sim g_n(t') + \mathcal{GP}(\mathbf{0}, \mathbf{K}_f(t,t')).
\end{aligned} \tag{31}$$

In this case we have a vector valued function $\mathbf{f}_{nr}$ at time t and at time t'. Notice that at same time $\mathbf{g}_n(t) \sim \mathcal{GP}(0, \mathbf{K}_g(t,t'))$. Thus the joint distribution of $\mathbf{f}_{nr}(t)$ and $\mathbf{f}_{nr}(t')$ is given by:

$$p(\mathbf{f}_{nr}(t), \mathbf{f}_{nr}(t')) = \mathcal{GP}(\mathbf{0}, \mathbf{K}_g) + \mathcal{GP}(\mathbf{0}, \mathbf{K}_f), \tag{32}$$

where $\mathbf{K}_g$ and $\mathbf{K}_f$ is the covariance matrix between the input points t and t' and across all the outputs. It is evident that the expected value of the joint distribution above is equal to the zero mean vector. Also is straightforward to see that the covariance of the joint distribution is given by $\mathbf{K}_g + \mathbf{K}_f$. Nevertheless one can show this property using the procedure below:

Defining the column vector $\hat{\mathbf{g}}_n \sim \mathcal{GP}(\mathbf{0}, \mathbf{K}_g)$ and the column vector $\hat{\mathbf{h}}_{nr} \sim \mathcal{GP}(\mathbf{0}, \mathbf{K}_f)$, $\mathbf{g}_n(t)$ as the first half of $\hat{\mathbf{g}}_n$, $\mathbf{g}_n(t')$ as the second half of $\hat{\mathbf{g}}_{\mathbf{n}}$, $\mathbf{h}_{nr}(t)$ as the first half of $\hat{\mathbf{h}}_{nr}$ and $\mathbf{h}_{nr}(t)$ as the second half of $\hat{\mathbf{h}}_{nr}$ the covariance of the joint distribution of $\mathbf{f}_{nr}(t)$ and $\mathbf{f}_{nr}(t')$ can be written as:

$$\begin{aligned}
\mathrm{cov}[\mathbf{f}_{nr}(t), \mathbf{f}_{nr}(t')] &= \mathrm{cov}[\mathbf{g}_n(t) + \mathbf{h}_{nr}(t), \mathbf{g}_n(t') + \mathbf{h}_{nr}(t')] \\
&= \mathrm{cov}[\mathbf{g}_n(t), \mathbf{g}_n(t') + \mathbf{h}_{nr}(t')] + \mathrm{cov}[\mathbf{h}_{nr}(t), \mathbf{g}_n(t') + \mathbf{h}_{nr}(t')] \\
&= [\mathrm{cov}[\mathbf{g}_n(t'), \mathbf{g}_n(t)] + \mathrm{cov}[\mathbf{h}_{nr}(t'), \mathbf{g}_n(t)]]^\top + [\mathrm{cov}[\mathbf{g}_n(t'), \mathbf{h}_{nr}(t)] + \mathrm{cov}[\mathbf{h}_{nr}(t'), \mathbf{h}_{nr}(t)]]^\top \\
&= \mathrm{cov}[\mathbf{g}_n(t'), \mathbf{g}_n(t)]^\top + \mathrm{cov}[\mathbf{h}_{nr}(t'), \mathbf{h}_{nr}(t)]^\top \\
&= \mathbf{K}_g(t, t')^\top + \mathbf{k}_f(t, t')^\top \\
&\text{For symmetry} \\
&= \mathbf{K}_g(t, t') + \mathbf{K}_f(t, t')
\end{aligned} \tag{33}$$

For the case $\mathbf{h}_{nr}(t')$ and $\mathbf{g}_n(t)$ are generated independently as well as $\mathbf{h}_{nr}(t)$ and $\mathbf{g}_n(t')$ so $\mathrm{cov}[\mathbf{h}_{nr}(t'), \mathbf{g}_n(t)] = \mathrm{cov}[\mathbf{g}_n(t'), \mathbf{h}_{nr}(t)] = 0$. It is straightforward to see that the mean of the joint distribution of two points from different replicates is equal to zero only changing the left side of equation (32) to $p(\mathbf{f}_{nr}(t), \mathbf{f}_{nr'}(t'))$, now the points are from different replicates. Furthermore the covariance can be derived using the same procedure above:

$$\begin{aligned}
\mathrm{cov}[\mathbf{f}_{nr}(t), \mathbf{f}_{nr'}(t')] &= \mathrm{cov}[\mathbf{g}_n(t) + \mathbf{h}_{nr}(t), \mathbf{g}_n(t') + \mathbf{h}_{nr'}(t')] \\
&= \mathrm{cov}[\mathbf{g}_n(t), \mathbf{g}_n(t') + \mathbf{h}_{nr'}(t')] + \mathrm{cov}[\mathbf{h}_{nr'}(t), \mathbf{g}_n(t') + \mathbf{h}_{nr'}(t')] \\
&= [\mathrm{cov}[\mathbf{g}_n(t'), \mathbf{g}_n(t)] + \mathrm{cov}[\mathbf{h}_{nr'}(t'), \mathbf{g}_n(t)]]^\top + [\mathrm{cov}[\mathbf{g}_n(t'), \mathbf{h}_{nr'}(t)] + \mathrm{cov}[\mathbf{h}_{nr'}(t'), \mathbf{h}_{nr}(t)]]^\top \\
&= [\mathrm{cov}[\mathbf{g}_n(t'), \mathbf{g}_n(t)]]^\top \\
&= \mathbf{K}_g(t, t')
\end{aligned} \tag{34}$$

For the case $\mathbf{h}_{nr'}(t')$ and $\mathbf{h}_{nr}(t)$ are generated independently, despite of the fact they have the same mean, thus $\mathrm{cov}[\mathbf{h}_{nr'}(t'), \mathbf{h}_{nr}(t)] = 0$.

A.3 Proposed Model: Gradients for the likelihood function (Two layer hierarchy)

According to the proposed model likelihood (see equation (18)) the derivative of the log likelihood w.r.t one of the parameters of the model is equal to

$$\frac{\partial \log p(\mathbf{Y}_n \mid \mathbf{T}_n, \boldsymbol{\phi})}{\partial \phi_i} = -\frac{1}{2} \hat{\mathbf{y}}_n^\top \boldsymbol{\Sigma}_n^{-1} \frac{\partial \boldsymbol{\Sigma}_n}{\partial \phi_i} \boldsymbol{\Sigma}_n^{-1} \hat{\mathbf{y}}_n - \frac{1}{2} \mathrm{tr}\left(\boldsymbol{\Sigma}_n^{-1} \frac{\partial \boldsymbol{\Sigma}_n}{\partial \phi_i} \right). \tag{35}$$

Thus the model likelihood derivative is reduced to compute the term $\frac{\partial \boldsymbol{\Sigma}_n}{\partial \phi_i}$ which is the derivative of the likelihood matrix w.r.t to the model parameters. The matrix derivative w.r.t to an scalar is equivalent to derive each element of the matrix w.r.t to the parameter ϕ_i in this case. However, to simplify the computations an alternative representation of the matrix $\mathbf{\Sigma}_n$ is given by equation (36).

$$\mathbf{\Sigma}_n = \mathbf{\Sigma}_{ng} + \mathbf{\Sigma}_{nf} + \mathbf{\Sigma}_{n\Gamma} \tag{36}$$

Where $\mathbf{\Sigma}_{ng}$ is a block covariance matrix, with same size as $\mathbf{\Sigma}_n$ in which all the entries of this matrix are equal to $\mathbf{K}_g(t,t')$, $\mathbf{\Sigma}_{nf}$ is a block covariance in which the block diagonal $r = r'$ is equal to $\mathbf{K}_f(t,t')$ all the other values left are equal to zero and finally $\mathbf{\Sigma}_{n\Gamma}$ is equal to a diagonal matrix containing the noise values $\{\beta_d\}_{d=1}^{D}$ for each one of the outputs of the observed child vector valued signals. In figure 25 there is an explanation of the equivalent representation derived before in equation (36).

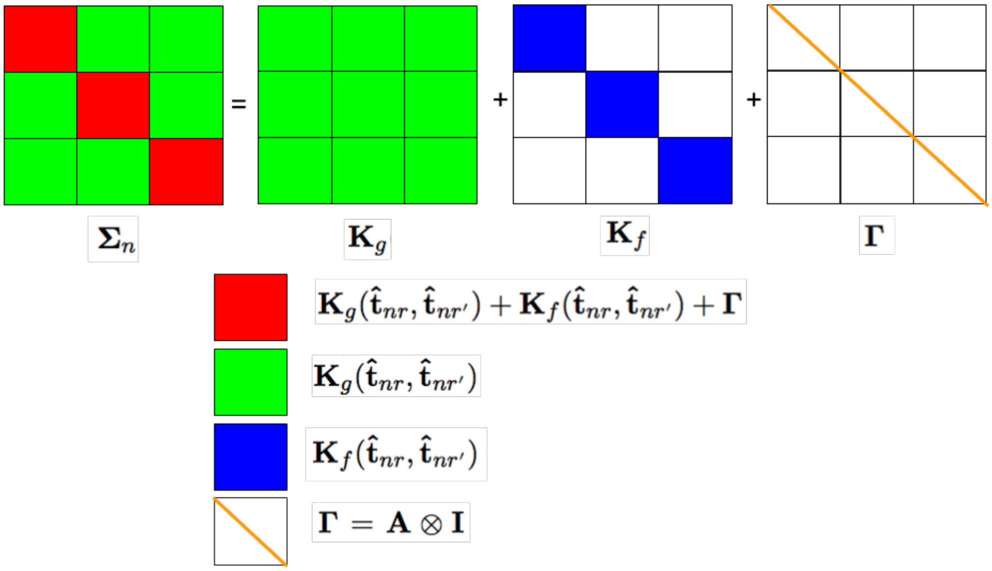

=
+
+
$\mathbf{\Sigma}_n$
$\mathbf{K}_g$
$\mathbf{K}_f$
$\mathbf{\Gamma}$
$\mathbf{K}_g(\hat{\mathbf{t}}_{nr}, \hat{\mathbf{t}}_{nr'}) + \mathbf{K}_f(\hat{\mathbf{t}}_{nr}, \hat{\mathbf{t}}_{nr'}) + \mathbf{\Gamma}$
$\mathbf{K}_g(\hat{\mathbf{t}}_{nr}, \hat{\mathbf{t}}_{nr'})$
$\mathbf{K}_f(\hat{\mathbf{t}}_{nr}, \hat{\mathbf{t}}_{nr'})$
$\mathbf{\Gamma} = \mathbf{A} \otimes \mathbf{I}$

Figure 25: Graphical view of the alternative representation for the proposed model likelihood matrix $\mathbf{\Sigma}_n$, in this case the blank spaces are equal to zero, $\mathbf{\Sigma}_n$ in this diagram can be thought as an simplification of the likelihood matrix specified in figure 3

.

Assuming there is only one covariance function for the input space, with its respective lenghtscale and variance hyperparameters one can simplify the model parameters in $\boldsymbol{\phi}$ as: the noise parameters $\{\beta_d\}_{d=1}^{D}$, the variance parameters σ_{gq} and σ_{fl} of k_{gq} and k_{fl} respectively, the lenghtscale parameters θ_{gq} and θ_{fl} and each one of the entries of the coregionalization matrices of each layer call them $b_{gq}^{dd'}$ and $b_{fl}^{dd'}$ corresponding to the $\mathbf{B}_{gq}$ and $\mathbf{B}_{fl}$ respectively. The derivative for some of the parameters will be further derived here assuming the altern representation for the likelihood matrix presented before. Thus, the derivative of the likelihood matrix w.r.t to one of the hyperparameters of k_{gq}, for example the variance σ_{gq}, is computed as follows:

$$\begin{aligned}\frac{\partial \boldsymbol{\Sigma}_n}{\partial \sigma_g q} &= \frac{\partial \boldsymbol{\Sigma}_{ng}}{\partial \sigma_{gq}} + \frac{\partial \boldsymbol{\Sigma}_{nf}}{\partial \sigma_{gq}} + \frac{\partial \boldsymbol{\Sigma}_{n\Gamma}}{\partial \sigma_{gq}} \\ &= \frac{\partial \boldsymbol{\Sigma}_{ng}}{\partial \sigma_{gq}} + \mathbf{0} + \mathbf{0},\end{aligned} \tag{37}$$

The only matrix here that has the interest terms σ_{gq} is the matrix $\boldsymbol{\Sigma}_{ng}$ that is why the other matrix derivatives are equal to $\mathbf{0}$. Thus, each element of the remaining matrix $\boldsymbol{\Sigma}_{ng}$ will be of the form $b_{gq}^{dd'} k_{gq}(t,t')$, so the derivative each element of $\frac{\partial \boldsymbol{\Sigma}_{ng}}{\partial \sigma_{gq}}$ will be of the form

$$b_{gq}^{dd'} \frac{\partial k_{gq}(t,t')}{\partial \sigma_{gq}}. \tag{38}$$

In the same way the derivatives w.r.t to one of the parameters of the corregionalization matrix $\mathbf{B}_{fl}$, in this case $b_{fl}^{dd'}$, will be computed as

$$\begin{aligned}\frac{\partial \boldsymbol{\Sigma}_n}{\partial b_{fl}^{dd'}} &= \frac{\partial \boldsymbol{\Sigma}_{ng}}{\partial b_{fl}^{dd'}} + \frac{\partial \boldsymbol{\Sigma}_{nf}}{\partial b_{fl}^{dd'}} + \frac{\partial \boldsymbol{\Sigma}_{n\Gamma}}{\partial b_{fl}^{dd'}} \\ &= \mathbf{0} + \frac{\partial \boldsymbol{\Sigma}_{nf}}{\partial b_{fl}^{dd'}} + \mathbf{0},\end{aligned} \tag{39}$$

the only matrix here with the interest terms is the second one. Thus, each element of the remaining matrix $\boldsymbol{\Sigma}_{nf}$ will be of the form $b_{fl}^{dd'} k_{fl}(t,t')$ in the block diagonal (see figure 25), so the derivative each element of $\frac{\partial \boldsymbol{\Sigma}_{nf}}{\partial \sigma_{fl}}$ will be of the form

$$\frac{\partial b_{fl}^{dd'} k_{fl}(t,t')}{\partial b_{fl}^{dd'}} = k_{fl}(t,t'). \tag{40}$$

It is quite straightforward to show that $\frac{\partial \boldsymbol{\Sigma}_n}{\partial \beta_d}$ is a zero valued matrix except to the elements on the diagonal containing β_d which will be equal to one. This process is repeated for all the parameters of the model until the gradient vector is completed, then it can be used in a gradient based optimization algorithm.

A.4 Discussion on synthesis evaluation measures

A.4.1 Discussion on synthesis measures over artificial data

Here we explain the process in which the reliability of the correlation measures was established regarding the tests with artificially generated training data. Intuitively, one will expect that the generated samples from the estimated models to be similar to the ones used to compare and created from the real process, which are different from the training ones. Thus, a control model was added, this model should be a different process from the one used to generate the training samples and the estimated ones (MOA and MOHGP). The samples from this control model should have a different correlation measure, in the case of the MCC to be lower than both MOA and MOHGP, and in the case of MLC to have a value closer to zero or even negative.

This control test makes more sense here since one is able to control the parameters of the generator process and the control test process. The things become different when the real MOCAP data is used. That will be the topic of the next section.

A.4.2 Discussion on synthesis measures over MOCAP data

The case of real MOCAP data is different from the case of artificial data. This is due to the fact that with artificial data one can have more control about the way in which the samples are generated. Thus, when comparing with the control test, the correlation measures seem to work in an acceptable way. However with the MOCAP data the things become difficult in the sense that the real MOCAP subjects used to compare the generated ones can be very different between each other. This difference can become in interpretations that initially can be counterintuitive if the measured thing here is the similarity.

To better explain this last statement let us take five MOCAP subjects performing a walking motion and five of them shooting a soccer ball. First, let us compute the correlation measures defined in this work for each one of the subjects against each other. In table 18 and 19 the results for the MCC (Mean Cross Correlation) and MLC (Mean Linear Correlation) are shown in the case of the walking subjects.

Table 18: MCC between walking evaluation subjects

Test subject	subj 12(1)	subj 16(15)	subj 38(2)	subj 35(1)	subj 39(4)
subj 12(1)	26.429	20.551	18.249	19.771	19.398
subj 16(15)	20.551	28.002	17.827	25.769	23.335
subj 38(2)	18.249	17.827	25.863	19.234	18.821
subj 35(1)	19.771	25.769	19.234	28.961	24.876
subj 39(4)	19.398	23.335	18.821	24.876	27.447

The diagonal in tables 18 and 19 shows the correlation measures of each subject against itself. Thus, it makes sense to have a linear correlation equal to one in table 19. On the other hand the diagonal in the table 18 can be interpreted as the desired value if one walking sample "wishes" to be similar to that particular subject. However, in both tables the values of similarity to subject 12(1), for example, are not that close to the desired value. These fact does not mean that the other subjects don't walk in a realistic manner (they are the real samples in first place), but rather that these subjects have different motion features. While subject 16(15) could be going faster, the subject 38(12) is very static, and that different features affects the correlation measures.

Table 19: MLC between walking evaluation subjects

Test subject	subj 12(1)	subj 16(15)	subj 38(2)	subj 35(1)	subj 39(4)
subj 12(1)	1.	-0.004	-0.160	-0.208	-0.087
subj 16(15)	-0.004	1.	-0.286	0.509	0.382
subj 38(2)	-0.160	-0.286	1.	-0.203	-0.192
subj 35(1)	-0.208	0.509	-0.203	1.	0.466
subj 39(4)	-0.087	0.382	-0.192	0.466	1.

For that particular reason the synthesis measures from tables 14, 15 and 16 have to be read but to take in to account the highest correlation value, given that this measure represents the subject which the generated motion samples are more similar to. After this observation one can compare both models (MOA and MOHGP) to see if the best value is better and determine which model was more accurate generating motion samples similar to the real ones.

However it is interesting also to check the subjects moving with a different kind of motion (soccer) against walking. This particular control test will show if the correlation measure is completely effective to evaluate the similarity between motions. In table 20 and 21 the results of comparing the soccer shooting subjects against the walking subjects are shown. One can conclude from these results that there are some subjects shooting a soccer ball more similar to some walking subjects than other walking subjects. For example, look the first row of table 20 all the soccer subjects are more similar to the walking subject 12(1) just because the correlation is closer to 26.249 which is the correlation of 12(1) against itself.

Table 20: MCC between Walking and soccer evaluation subjects.

Test subject	subj 10(5)	subj 10(3)	subj 10(2)	subj 10(6)	subj 11(1)
subj 12(1)	22.171	22.270	22.042	23.116	21.667
subj 16(15)	21.842	22.141	22.369	23.212	21.656
subj 38(2)	20.262	22.403	20.550	23.551	22.173
subj 35(1)	20.949	22.070	20.978	22.704	21.164
subj 39(4)	21.884	21.790	22.168	22.523	21.528

Table 21: MLC between walking and soccer evaluation subjects.

Test subject	subj 10(5)	subj 10(3)	subj 10(2)	subj 10(6)	subj 11(1)
subj 12(1)	0.039	-0.072	0.094	-0.089	-0.100
subj 16(15)	0.004	-0.087	0.144	-0.072	-0.113
subj 38(2)	-0.081	0.099	-0.046	0.144	0.142
subj 35(1)	-0.016	0.006	0.048	-0.019	-0.029
subj 39(4)	0.066	-0.057	0.140	-0.103	-0.067

For that particular latter reason, the correlation measures can not be determined as an absolute way to determine the similarity of the motion samples artificially generated and the real ones. However, this issue is partially addressed if the models are compared against the best matches as explained before. The reason for some soccer shooting subjects being more similar than other walking subjects could be that there are similarities within the angles of each joints performing certain poses which leads to the correlations measures to be higher between some walking samples and some soccer samples. However further study about better methods to measure the similarity between MOCAP subjects should be done.

Bibliography

[1] J. Hensman, N. D. Lawrence, and M. Rattray, "Hierarchical Bayesian modelling of gene expression time series across irregularly sampled replicates and clusters," *BMC Bioinformatics*, vol. 14, no. 1, pp. 1–12, 2013. [Online]. Available: http://dx.doi.org/10.1186/1471-2105-14-252

[2] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian Data Analysis, Third Edition.* 6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742.: CRC Press, Taylor & Francis Group, 2014.

[3] J. Shi, R. Murray-Smith, and D. Titterington, "Hierarchical Gaussian Process mixtures for regression," *Statistics and Computing*, vol. 15, no. 1, pp. 31–41, 2005. [Online]. Available: http://dx.doi.org/10.1007/s11222-005-4787-7

[4] M. W. Wheeler, D. B. Dunson, S. P. Pandalai, B. A. Baker, and A. H. Herring, "Mechanistic hierarchical gaussian processes," *Journal of the American Statistical Association*, vol. 109, no. 507, pp. 894–904, 2014. [Online]. Available: http://dx.doi.org/10.1080/01621459.2014.899234

[5] S. Park and S. Choi, "Hierarchical Gaussian Process Regression," in *Proceedings of 2nd Asian Conference on Machine Learning (ACML2010)*, vol. 13, 2009, pp. 95–110.

[6] Q. C. Xin Wang and W. Wang, "3d human motion editing and synthesis: A survey," *Computational and Mathematical Methods in Medicine*, vol. 2014, p. 11, 2014.

[7] L. Zhou, L. Shang, H. P. Shum, and H. Leung, "Human motion variation synthesis with multivariate Gaussian Processes," *Computer Animation and Virtual Worlds*, vol. 25, no. 3-4, pp. 301–309, 2014. [Online]. Available: http://dx.doi.org/10.1002/cav.1599

[8] N. D. Lawrence and A. J. Moore, "Hierarchical Gaussian Process Latent Variable Models," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 481–488. [Online]. Available: http://doi.acm.org/10.1145/1273496.1273557

[9] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Multifactor Gaussian Process Models for Style-content Separation," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 975–982. [Online]. Available: http://doi.acm.org/10.1145/1273496.1273619

[10] S. Fukayama and M. Goto, "Automated Choreography Synthesis Using a Gaussian Process Leveraging Consumer-generated Dance Motions," in *Proceedings of the 11th Conference on Advances in Computer Entertainment Technology*, ser. ACE

'14. New York, NY, USA: ACM, 2014, pp. 23:1–23:6. [Online]. Available: http://doi.acm.org/10.1145/2663806.2663849

[11] N. Taubert, A. Christensen, D. Endres, and M. A. Giese, "Online Simulation of Emotional Interactive Behaviors with Hierarchical Gaussian Process Dynamical Models," in *Proceedings of the ACM Symposium on Applied Perception*, ser. SAP '12. New York, NY, USA: ACM, 2012, pp. 25–32. [Online]. Available: http://doi.acm.org/10.1145/2338676.2338682

[12] N. Courty and A. Cuzol, "Conditional Stochastic Simulation for Character Animation," *Computer Animation and Virtual Worlds (best selected papers from CASA 2010)*, pp. 1–10, 2010. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00493340

[13] C. Guo, S. Ruan, and X. Liang, "Synthesis and editing of human motion with generative human motion model," in *2015 International Conference on Virtual Reality and Visualization (ICVRV)*, Oct 2015, pp. 193–196.

[14] L. Ikemoto, O. Arikan, and D. Forsyth, "Generalizing Motion Edits with Gaussian Processes," *ACM Trans. Graph.*, vol. 28, no. 1, pp. 1:1–1:12, Feb. 2009. [Online]. Available: http://doi.acm.org/10.1145/1477926.1477927

[15] M. A. Álvarez, L. Rosasco, and N. D. Lawrence, "Kernels for vector-valued functions: A review," *Found. Trends Mach. Learn.*, vol. 4, no. 3, pp. 195–266, Mar. 2012. [Online]. Available: http://dx.doi.org/10.1561/2200000036

[16] A. Schwaighofer, V. Tresp, and K. Yu, "Learning Gaussian Process Kernels via Hierarchical Bayes," in *Advances in Neural Information Processing Systems 17*. MIT Press, 2005. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=74492

[17] K. Yamane, Y. Ariki, and J. Hodgins, "Animating non-humanoid characters with human motion data," in *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '10. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2010, pp. 169–178. [Online]. Available: http://dl.acm.org/citation.cfm?id=1921427.1921453

[18] K. Yamane, M. Revfi, and T. Asfour, "Synthesizing object receiving motions of humanoid robots with human motion database," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 1629–1636.

[19] K. Perlin and A. Goldberg, "Improv: A system for scripting interactive actors in virtual worlds," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '96. New York, NY, USA: ACM, 1996, pp. 205–216. [Online]. Available: http://doi.acm.org/10.1145/237170.237258

[20] D. Chi, M. Costa, L. Zhao, and N. Badler, "The emote model for effort and shape," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '00. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 173–182. [Online]. Available: http://dx.doi.org/10.1145/344779.352172

[21] J. Starck, G. Miller, and A. Hilton, "Video-based character animation," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '05. New York, NY, USA: ACM, 2005, pp. 49–58. [Online]. Available: http://doi.acm.org/10.1145/1073368.1073375

[22] W. Lee, J. Gu, and N. Magnenat-Thalmann, "Generating animatable 3d virtual humans from photographs," *Computer Graphics Forum*, vol. 19, no. 3, pp. 1–10, 2000. [Online]. Available: http://dx.doi.org/10.1111/1467-8659.00392

[23] F. J. Perales and J. Torres, "A system for human motion matching between synthetic and real images based on a biomechanic graphical model," in *Motion of Non-Rigid and Articulated Objects, 1994., Proceedings of the 1994 IEEE Workshop on*, Nov 1994, pp. 83–88.

[24] A. Elhayek, E. Aguiar, A. Jain, J. Tompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, and C. Theobalt, "Efficient convnet-based marker-less motion capture in general scenes with a low number of cameras," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[25] X. Wei and J. Chai, "Videomocap: Modeling physically realistic human motion from monocular video sequences," in *ACM SIGGRAPH 2010 Papers*, ser. SIGGRAPH '10. New York, NY, USA: ACM, 2010, pp. 42:1–42:10. [Online]. Available: http://doi.acm.org/10.1145/1833349.1778779

[26] N. R. Howe, M. E. Leventon, and W. T. Freeman, "Bayesian reconstruction of 3d human motion from single-camera video," in *Advances in Neural Information Processing Systems*. MIT Press, 1999, pp. 820–826.

[27] A. C. Fang and N. S. Pollard, "Efficient synthesis of physically valid human motion," in *ACM SIGGRAPH 2003 Papers*, ser. SIGGRAPH '03. New York, NY, USA: ACM, 2003, pp. 417–426. [Online]. Available: http://doi.acm.org/10.1145/1201775.882286

[28] J. K. Hodgins, W. L. Wooten, D. C. Brogan, and J. F. O'Brien, "Animating human athletics," in *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '95. New York, NY, USA: ACM, 1995, pp. 71–78. [Online]. Available: http://doi.acm.org/10.1145/218380.218414

[29] S. Jain, Y. Ye, and C. K. Liu, "Optimization-based interactive motion synthesis," *ACM Trans. Graph.*, vol. 28, no. 1, pp. 10:1–10:12, Feb. 2009. [Online]. Available: http://doi.acm.org/10.1145/1477926.1477936

[30] C. K. Liu, A. Hertzmann, and Z. Popović, "Learning physics-based motion style with nonlinear inverse optimization," in *ACM SIGGRAPH 2005 Papers*, ser. SIGGRAPH '05. New York, NY, USA: ACM, 2005, pp. 1071–1081. [Online]. Available: http://doi.acm.org/10.1145/1186822.1073314

[31] J. McCann, N. S. Pollard, and S. Srinivasa, "Physics-based motion retiming," in *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '06. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2006, pp. 205–214. [Online]. Available: http://dl.acm.org/citation.cfm?id=1218064.1218092

[32] J. Min, Y.-L. Chen, and J. Chai, "Interactive generation of human animation with deformable motion models," *ACM Trans. Graph.*, vol. 29, no. 1, pp. 9:1–9:12, Dec. 2009. [Online]. Available: http://doi.acm.org/10.1145/1640443.1640452

[33] L. Kovar, M. Gleicher, and F. Pighin, "Motion Graphs," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 473–482, Jul. 2002. [Online]. Available: http://doi.acm.org/10.1145/566654.566605

[34] J. Lee, J. Chai, P. S. A. Reitsma, J. K. Hodgins, and N. S. Pollard, "Interactive control of avatars animated with human motion data," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 491–500, Jul. 2002. [Online]. Available: http://doi.acm.org/10.1145/566654.566607

[35] M. Mizuguchi, J. Buchanan, and T. Calvert, "Data driven motion transitions for interactive games," in *Eurographics 2001 - Short Presentations*. Eurographics Association, 2001.

[36] T. Kwon, K. H. Lee, J. Lee, and S. Takahashi, "Group motion editing," in *ACM SIGGRAPH 2008 Papers*, ser. SIGGRAPH '08. New York, NY, USA: ACM, 2008, pp. 80:1–80:8. [Online]. Available: http://doi.acm.org/10.1145/1399504.1360679

[37] M. Brand and A. Hertzmann, "Style Machines," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '00. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 183–192. [Online]. Available: http://dx.doi.org/10.1145/344779.344865

[38] L. M. Tanco and A. Hilton, "Realistic synthesis of novel human movements from a database of motion capture examples," in *Proceedings of the Workshop on Human Motion (HUMO'00)*, ser. HUMO '00. Washington, DC, USA: IEEE Computer Society, 2000, pp. 137–. [Online]. Available: http://dl.acm.org/citation.cfm?id=822088.823423

[39] Y. Li, T. Wang, and H.-Y. Shum, "Motion texture: A two-level statistical model for character motion synthesis," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 465–472, Jul. 2002. [Online]. Available: http://doi.acm.org/10.1145/566654.566604

[40] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian Process Dynamical Models for Human Motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 283–298, Feb 2008.

[41] L. Zhou, L. Shang, H. P. Shum, and H. Leung, "Human motion variation synthesis with multivariate Gaussian Processes," *Computer Animation and Virtual Worlds*, vol. 25, no. 3-4, pp. 301–309, 2014. [Online]. Available: http://dx.doi.org/10.1002/cav.1599

[42] K. Pullen and C. Bregler, "Animating by multi-level sampling," in *Computer Animation 2000. Proceedings*, 2000, pp. 36–42.

[43] X. Wei, J. Min, and J. Chai, "Physically valid statistical models for human motion generation," *ACM Trans. Graph.*, vol. 30, no. 3, pp. 19:1–19:10, May 2011. [Online]. Available: http://doi.acm.org/10.1145/1966394.1966398

[44] C.-S. Lee and A. Elgammal, "Human motion synthesis by motion manifold learning and motion primitive segmentation," in *Proceedings of the 4th International Conference on Articulated Motion and Deformable Objects*, ser. AMDO'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 464–473. [Online]. Available: http://dx.doi.org/10.1007/11789239_48

[45] Y. Wang, J. Xiao, and B. Wei, "3d human motion synthesis based on nonlinear manifold learning," *Journal of Image and Graphics*, vol. 15, no. 6, pp. 936–943, 2010.

[46] J. Chai and J. K. Hodgins, "Performance animation from low-dimensional control signals," in *ACM SIGGRAPH 2005 Papers*, ser. SIGGRAPH '05. New York, NY, USA: ACM, 2005, pp. 686–696. [Online]. Available: http://doi.acm.org/10.1145/1186822.1073248

[47] R. Bowden, "Learning statistical models of human motion," 2000.

[48] P. Faloutsos, M. van de Panne, and D. Terzopoulos, "Composable controllers for physics-based character animation," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '01. New York, NY, USA: ACM, 2001, pp. 251–260. [Online]. Available: http://doi.acm.org/10.1145/383259.383287

[49] M. A. Alvarez, D. Luengo, and N. D. Lawrence, "Linear Latent Force Models Using Gaussian Processes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2693–2705, Nov. 2013. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2013.86

[50] V. B. Zordan and J. K. Hodgins, *Tracking and Modifying Upper-body Human Motion Data with Dynamic Simulation.* Vienna: Springer Vienna, 1999, pp. 13–22. [Online]. Available: http://dx.doi.org/10.1007/978-3-7091-6423-5_2

[51] O. C. Jenkins and M. J. Mataric, "Deriving action and behavior primitives from human motion," in *In International Conference on Intelligent Robots and Systems*, 2002, pp. 2551–2556.

[52] K. Yamane, J. J. Kuffner, and J. K. Hodgins, "Synthesizing animations of human manipulation tasks," in *ACM SIGGRAPH 2004 Papers*, ser. SIGGRAPH '04. New York, NY, USA: ACM, 2004, pp. 532–539. [Online]. Available: http://doi.acm.org/10.1145/1186562.1015756

[53] E. Hsu, K. Pulli, and J. Popović, "Style translation for human motion," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 1082–1089, Jul. 2005. [Online]. Available: http://doi.acm.org/10.1145/1073204.1073315

[54] G. Liu and L. McMillan, "Estimation of missing markers in human motion capture," *The Visual Computer*, vol. 22, no. 9, pp. 721–728, 2006. [Online]. Available: http://dx.doi.org/10.1007/s00371-006-0080-9

[55] D. J. Wiley and J. K. Hahn, "Interpolation synthesis of articulated figure motion," *IEEE Comput. Graph. Appl.*, vol. 17, no. 6, pp. 39–45, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1109/38.626968

[56] Z. Chen, L. Ma, Y. Gao, and X. Wu, "Human motion interpolation using space-time registration," in *Proceedings of the 16th International Conference on Advances in Artificial Reality and Tele-Existence*, ser. ICAT'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 38–47. [Online]. Available: http://dx.doi.org/10.1007/11941354_5

[57] A. Safonova and J. K. Hodgins, "Analyzing the physical correctness of interpolated human motion," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '05. New York, NY, USA: ACM, 2005, pp. 171–180. [Online]. Available: http://doi.acm.org/10.1145/1073368.1073392

[58] C. Rose, M. F. Cohen, and B. Bodenheimer, "Verbs and adverbs: Multidimensional motion interpolation," *IEEE Comput. Graph. Appl.*, vol. 18, no. 5, pp. 32–40, Sep. 1998. [Online]. Available: http://dx.doi.org/10.1109/38.708559

[59] A. Bruderlin and L. Williams, "Motion signal processing," in *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '95. New York, NY, USA: ACM, 1995, pp. 97–104. [Online]. Available: http://doi.acm.org/10.1145/218380.218421

[60] H. Mori and J. Hoshino, “Ica-based interpolation of human motion,” in *Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation. Computational Intelligence in Robotics and Automation for the New Millennium (Cat. No.03EX694)*, vol. 1, July 2003, pp. 453–458 vol.1.

[61] L. S. Brotman and A. N. Netravali, “Motion interpolation by optimal control,” in *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’88. New York, NY, USA: ACM, 1988, pp. 309–315. [Online]. Available: http://doi.acm.org/10.1145/54852.378531

[62] T. Mukai and S. Kuriyama, “Geostatistical motion interpolation,” in *ACM SIGGRAPH 2005 Papers*, ser. SIGGRAPH ’05. New York, NY, USA: ACM, 2005, pp. 1062–1070. [Online]. Available: http://doi.acm.org/10.1145/1186822.1073313

[63] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popović, “Style-based inverse kinematics,” in *ACM SIGGRAPH 2004 Papers*, ser. SIGGRAPH ’04. New York, NY, USA: ACM, 2004, pp. 522–531. [Online]. Available: http://doi.acm.org/10.1145/1186562.1015755