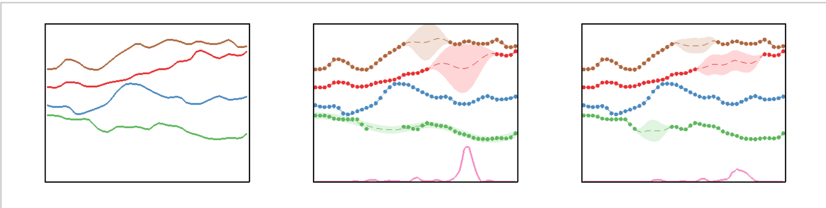
MTGP - A Multi-task Gaussian Process Toolbox



Authors

Robert Dürichen (University of Lübeck)
Marco AF Pimentel (University of Oxford)
Lei Clifton (University of Oxford)
Achim Schweikard (University of Lübeck)
David A Clifton (University of Oxford)

Abstract

Gaussian process (GP) models are a flexible means of performing non-parametric Bayesian regression. However, the majority of existing work using GP models in healthcare data is defined for univariate output time-series, denoted as single-task GPs (STGP). Here, we investigate how GPs could be used to model multiple correlated univariate physiological time-series simultaneously. The resulting multi-task GP (MTGP) framework can learn the correlation within multiple signals even though they might be sampled at different frequencies and have training sets available for different intervals. We illustrate the basic properties of MTGPs using a synthetic case-study with respiratory motion data. Finally, two real-world biomedical problems are investigated from the field of patient monitoring and motion compensation in radiotherapy. The results are compared to STGPs and other standard methods in the respective fields. In both cases, MTGPs learned the correlation between physiological time-series efficiently, which leads to improved modelling accuracy.

Literature

Dürichen, R., Pimentel, M.A.F., Clifton, L., Schweikard, A., and Clifton, D.A.:
Multi-task Gaussian Process Models for Biomedical Applications
IEEE Biomedical & Health Informatics, Valencia, Spain, 2014, pp. 492-495 [PDF]

Dürichen, R., Wissel, T., Ernst, F., Pimentel, M.A.F., Clifton, D.A., and Schweikard, A.:
Unified Approach for Respiratory Motion Prediction and Correlation with Multi-task Gaussian Processes
IEEE MLSP, Reims, France, 2014, pp. 1-6 [PDF]

Dürichen, R., Pimentel, M.A.F., Clifton, L., Schweikard, A., and Clifton, D.A.:
Multi-task Gaussian Processes for Multivariate Physiological Time-Series Analysis
IEEE Transactions on Biomedical Engineering 62(1), 2015, pp. 314 - 322 [PDF]

Download

The current release is v1.4: download "MTGP" toolbox for Matlab

As well as downloading the MTGP toolbox, you will need:
[GPML] - v3.4 or above; a Matlab toolbox for Gaussian processes.

Using this toolbox should be straightforward: the download comes with some toy datasets on which the demos, shown below can be run. Alternatively, use the scripts in the "example" folder to perform the same. Please feel free to contact the authors for more details concerning any of the scripts, or if you encounter any difficulties in using the toolbox.

Demos

[Demo 1]: Prediction of multiple correlated tasks with individual training data

[Demo 2]: Temporal shift between tasks

[Demo 3]: Tasks with different temporal characteristics

Group Pages

Home

News

People

Projects

Publications

Code

Lab Meetings

Contact

Publications...

Journals and Books

Conferences

Posters

Patents

Other

[Demo 4]: Interpreting the correlation coefficients

[Demo 5]: Noise free correlation analysis

Copyright

Acknowledgements

National Institute for Health Research

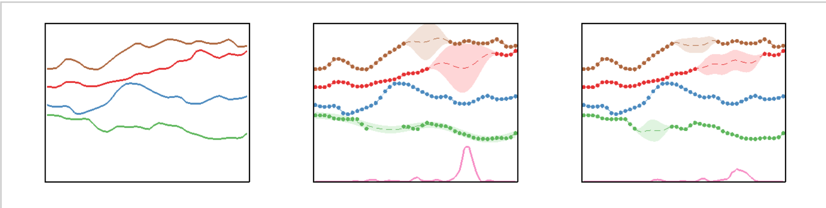
Healthcare Innovation Centre for Doctoral Training

FCT Fundação para a Ciência e a Tecnologia

Universität zu Lübeck Graduate School for Computing in Medicine and Life Sciences

 Updated January, 2015

MTGP - A Multi-task Gaussian Process Toolbox



Demo 1

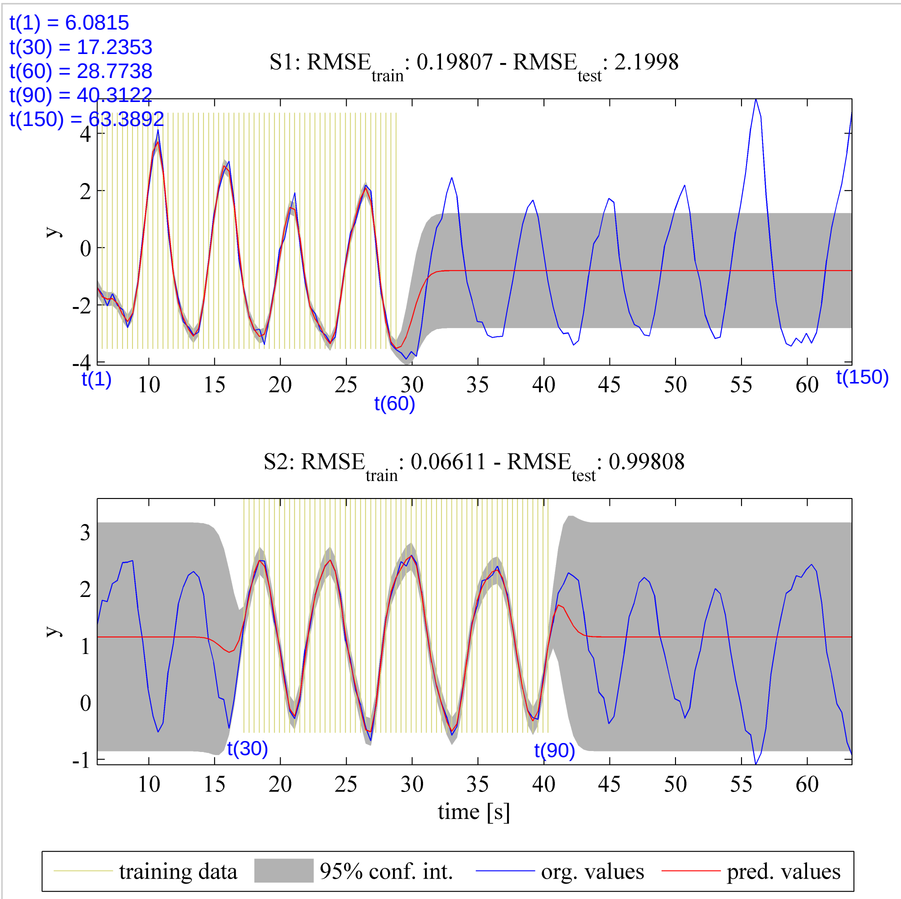
Prediction of multiple correlated tasks with individual training data

File name: *demoMTGP.m*

GP vs. MTGP

This is the first example which illustrates the differences between regression tasks using normal GPs and MTGPs. To use *demoMTGP.m*, please include the path of the GPML toolbox or specify the path in *demoMTGP.m* (line 6). The example contains 5 cases. The first case is the simplest case, which will be sequentially extended by each following case by certain aspects. Please select one of the cases using the variable *MTGP_case* (line 24).

line 24: **MTGP_case = 1** % default method



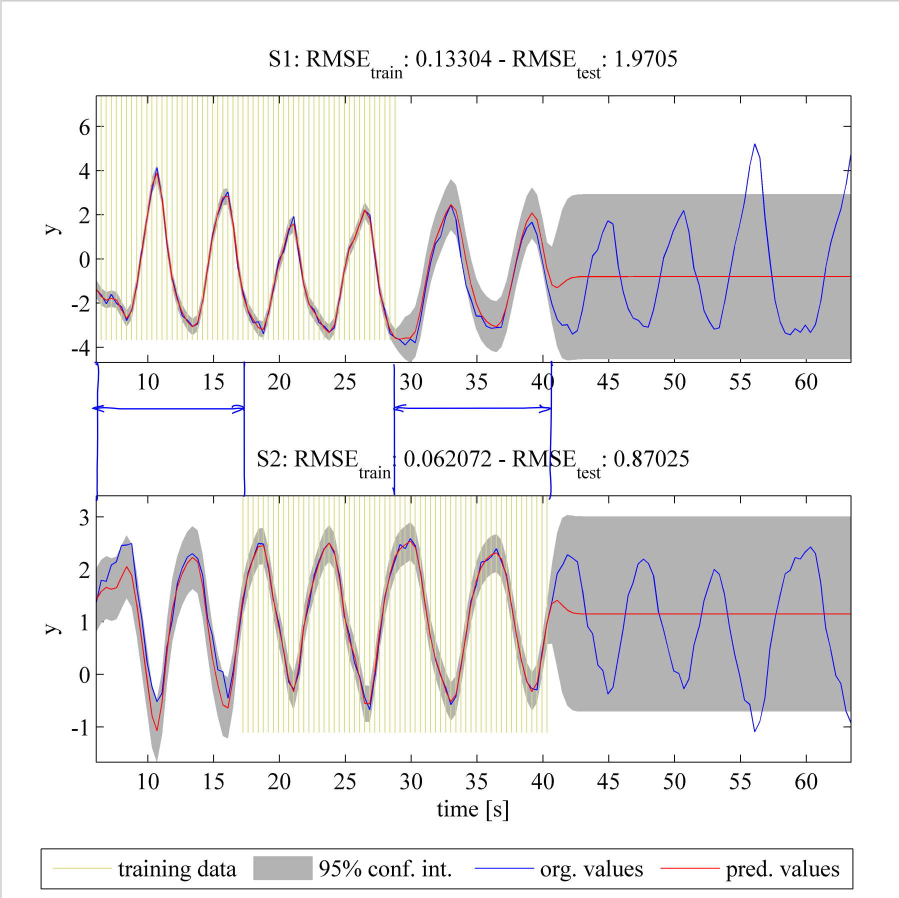
The plots above show the original "ground truth" dataset (blue solid line) for two periodical tasks (S1 and S2). The shaded yellow regions indicate the known training data for each task. For example, in the upper plot, we assume that data in the region t < 28 are available as training data. The results are generated by the MTGP toolbox using a squared exponential temporal covariance function K_t, assuming that the tasks are independent of each other. The hyperparameters of the covariance function were not optimised. The prediction values are shown with a red solid line (for the mean function) and the 95% confidence interval with the shaded grey area. The root mean square error

$$x_{rms} = \sqrt{\frac{1}{n}(x_1^2 + x_2^2 + \cdots + x_n^2)}$$

(RMSE) of the predictions compared with the training and test data for each task is shown above the plots.

The predictions shown in the two plots above can be interpreted as being those from using a single-task GP model on each time-series independently (in which correlation between tasks is not considered). Even though the tasks have different known training datasets, the prediction for the test data of each task is mainly the mean of the training data, as independence between the tasks is assumed. case1: 2 signals, assuming that they are uncorrelated, no further optimization is done and k_t = SE cov. func

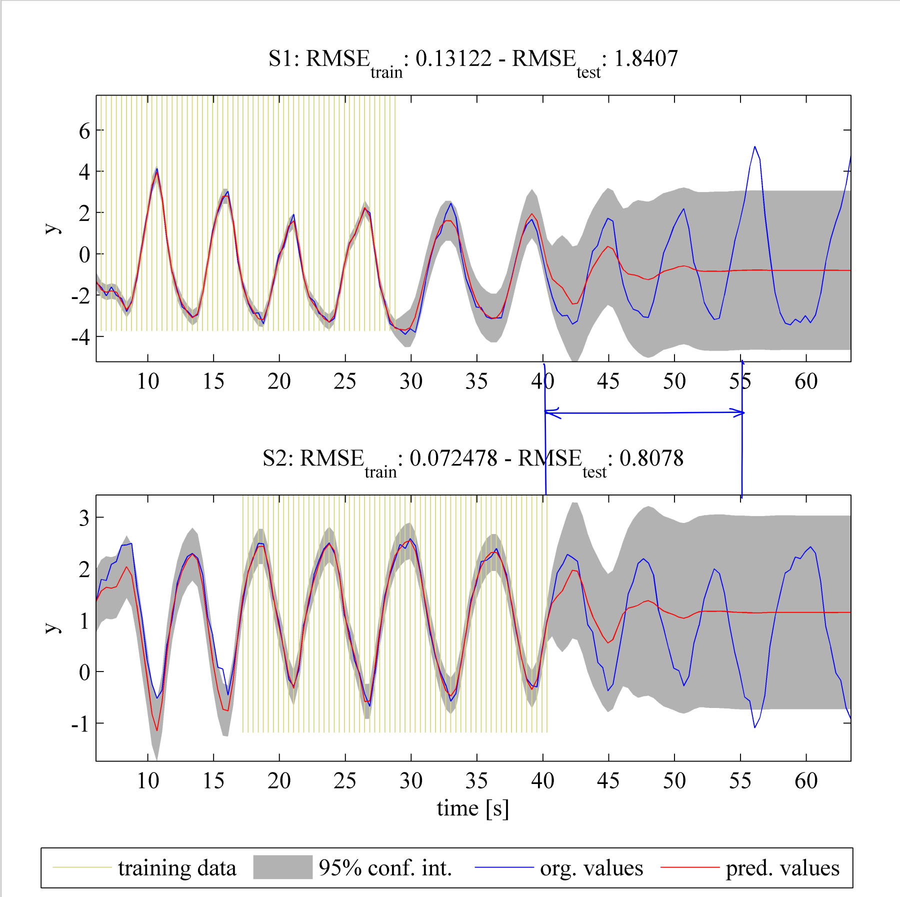
line 24: **MTGP_case = 2**



This case is an extension of case 1 using the same tasks, training data, and covariance function as before. However, in contrast to case 1, the MTGP model is initialised assuming independence between the tasks and then the hyperparameters of the GP are optimised by minimising the negative log marginal likelihood (NLML). The plots above again show the original dataset with the blue line, the training data as the shaded yellow region, and the predicted values as a red solid line and with the 95% confidence interval as shaded gray area. Compared to case 1, the MTGP model learns the negative correlation that exists between the tasks. NLML This information improves the predicted results for task S1 (shown in the upper plot) in the range of t = [30 40]s. Similarly, improved predictions may be seen for task S2 (shown in the lower plot) in the range t = [0 15]s. We can see that there is also a decrease in RMSE_test for both tasks, when compared with the RMSE_test values from case 1. case2: 2 signals, assuming that they are uncorrelated but with further optimization and k_t = SE cov. func

minimize(NLML)

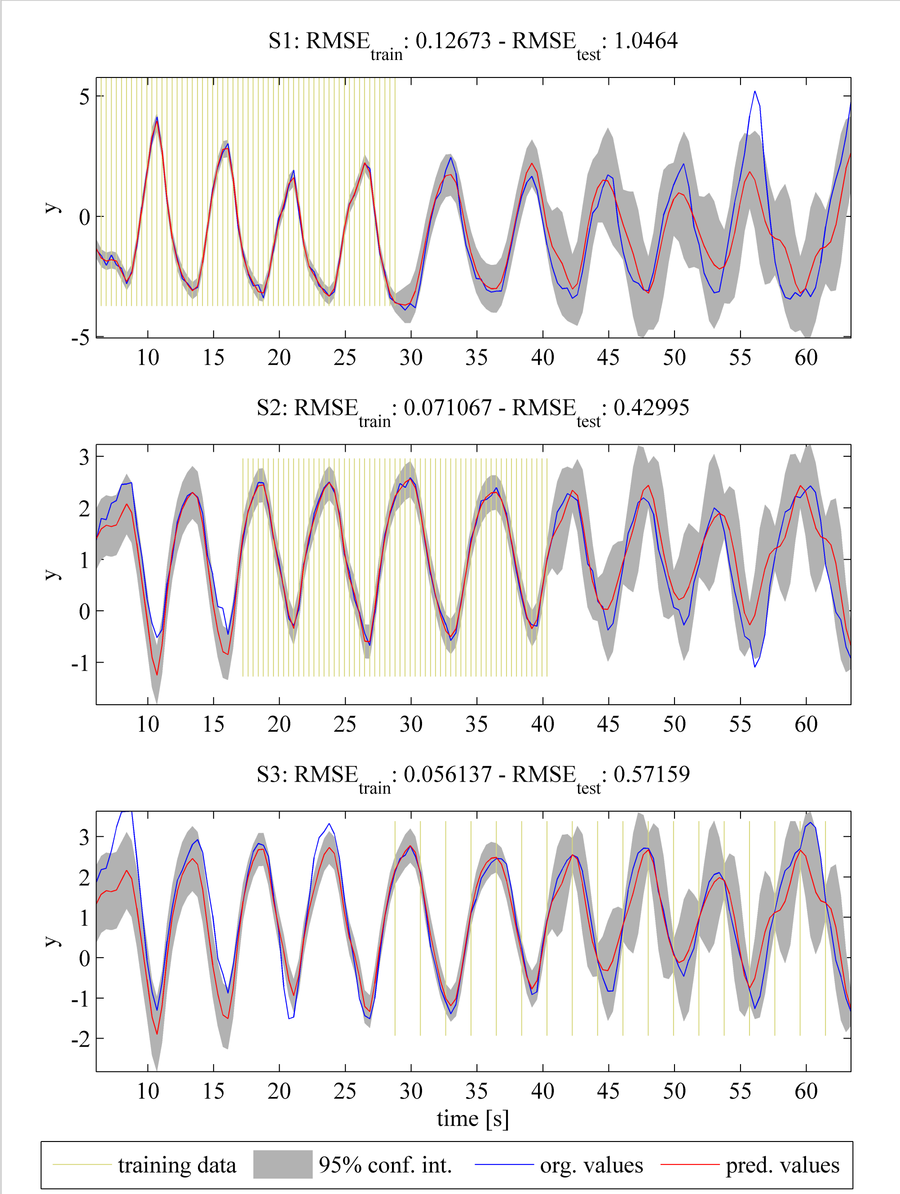
line 24: **MTGP_case = 3**



Case 3 demonstrates the use of prior knowledge of the functional behaviour (periodicity) in the MTGP model. Compared to case 2, the temporal covariance function is changed from a squared exponential to a quasi-periodic function (i.e., the multiplication of a squared exponential and a periodic covariance function). The remaining hyperparameters remain unchanged. Using prior knowledge of quasi-periodic functions in this way, improvements of the prediction performance may be seen for both task S1 and S2 in the range t = [40 55]s, with a corresponding further decrease in the values of RMSE_test for both tasks, when compared with the RMSE_test from previous cases.

case3: 2 signals, assuming that they are uncorrelated but with further optimization and K_t = quasi-periodic cov. func

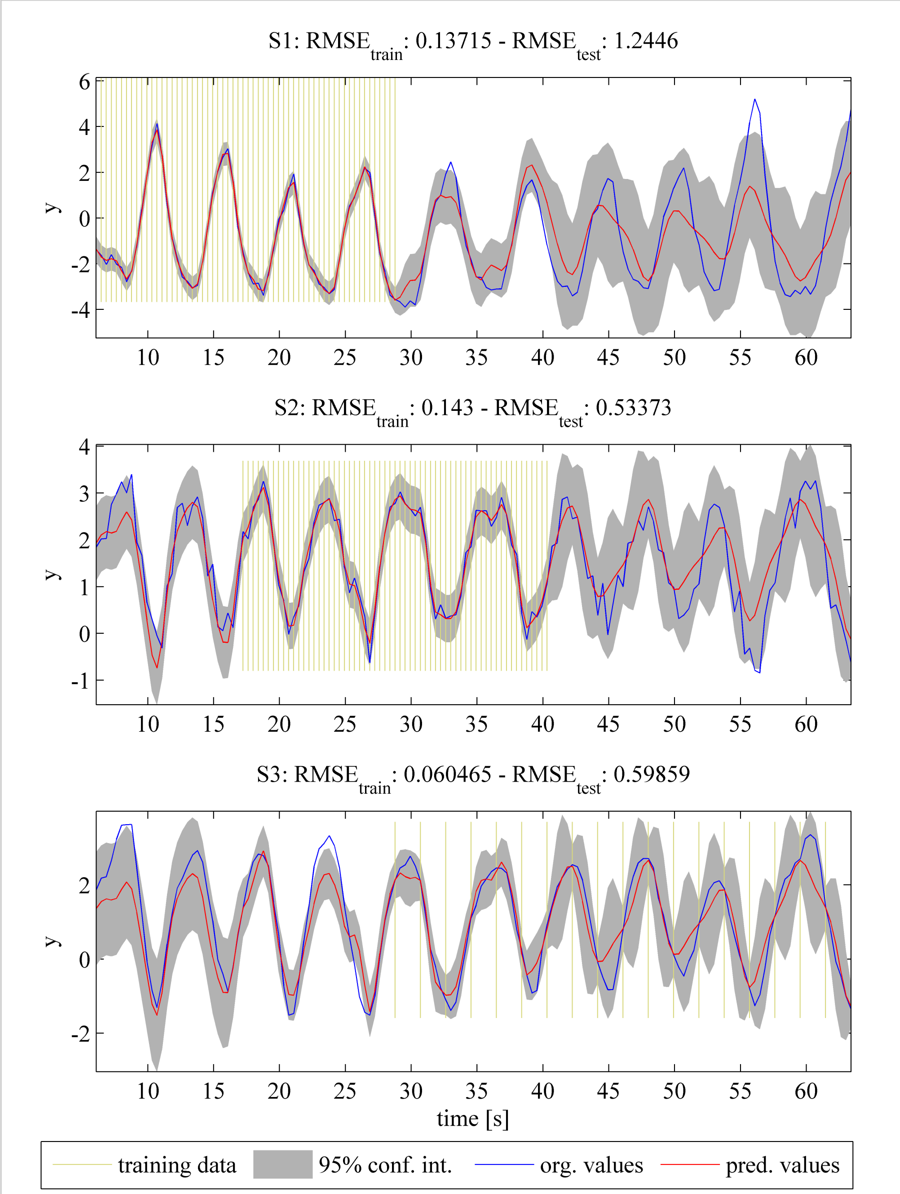
line 24: **MTGP_case = 4**



MTGP models can be extended to an arbitrary number of tasks. Assuming that a third task S3 is given, the task can be easily integrated into the MTGP model. We note that task S3 has a different sampling frequency - the MTGP model can cope with tasks of differing sampling frequencies in a straightforward manner. Similar to cases 2 and 3, considered previously, the model is here initialised assuming that all tasks are independent of each other. By optimising the NLML again, the MTGP model learns the correlation between tasks. We can see that inclusion of task S3 results in a further improvement in the prediction accuracies of tasks S1 and S2, with further decreases in RMSE_test compared with that from previous cases.

case4: 3 signals, assuming that all are uncorrelated and with different sample frequencies - k_t = quasi-periodic cov.func

line 24: **MTGP_case = 5**



opt.training_data{1} = 1:60; % 60
opt.training_data{2} = 30:90; % 61
opt.training_data{3} = 60:5:150;% 19

The final case in demo 1 illustrates how MTGP models can be extended to model task-specific noise. (In all previous cases, the noise was assumed to be the same for each task.) To show this, an additional noise component has been added to the original data from task S2. The temporal covariance function is now extended from a quasi-periodic function to a sum of a quasi-periodic and a noise covariance function. The noise covariance function adds one hyperparameter for each task expressing the noise for each task. The console displays the values of the hyperparameters learned for these GPs, and, as expected, it may be seen that the second hyperparameter, corresponding to the noise of task S2, takes a higher value compared with those for tasks S1 and S3.

3 signals, assuming that all are uncorrelated, with optimization and different sample frequencies

Return to MTGP toolbox page...

- noise is added to signal 2 - quasi-periodic cov. func with individual noise term for each signal.

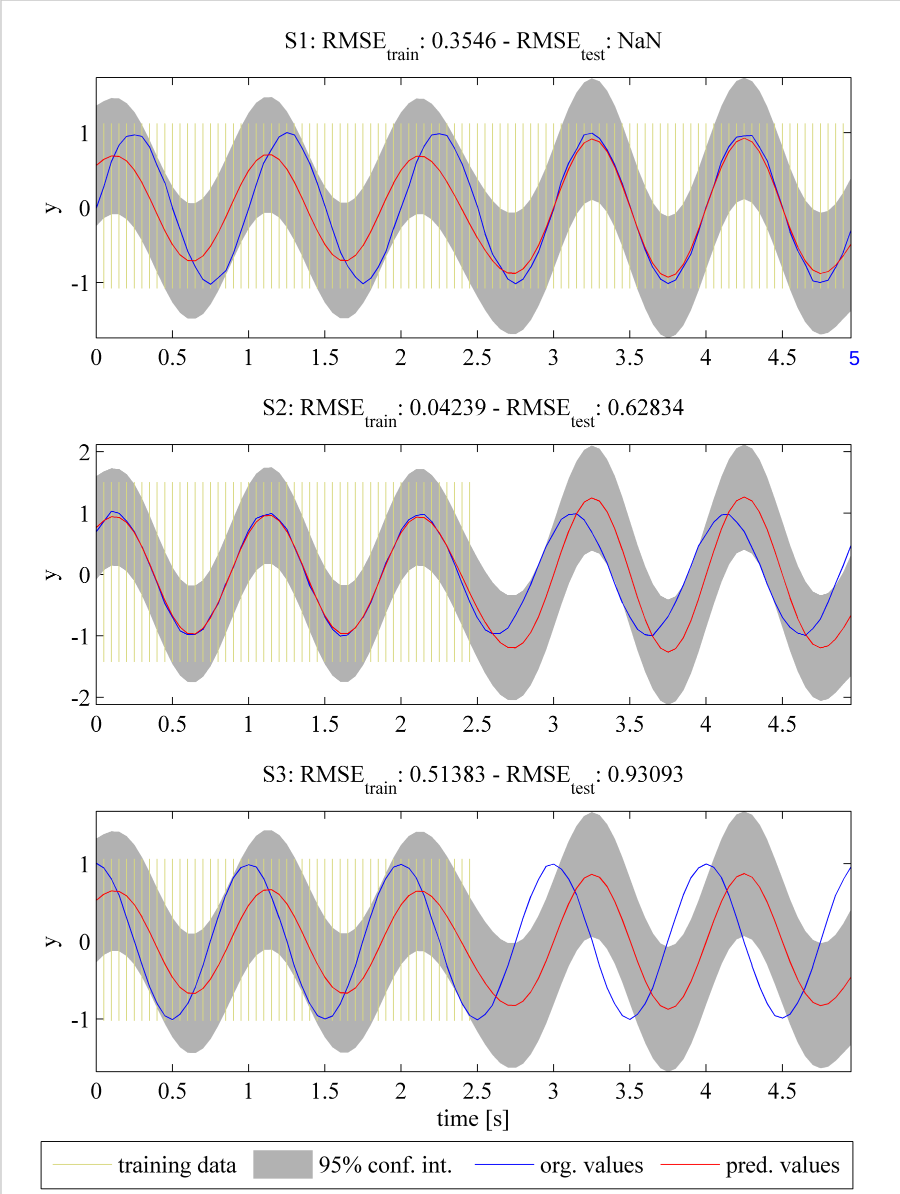
Demo 2

Temporal shift between tasks

File name: *demoMTGP_shift.m*

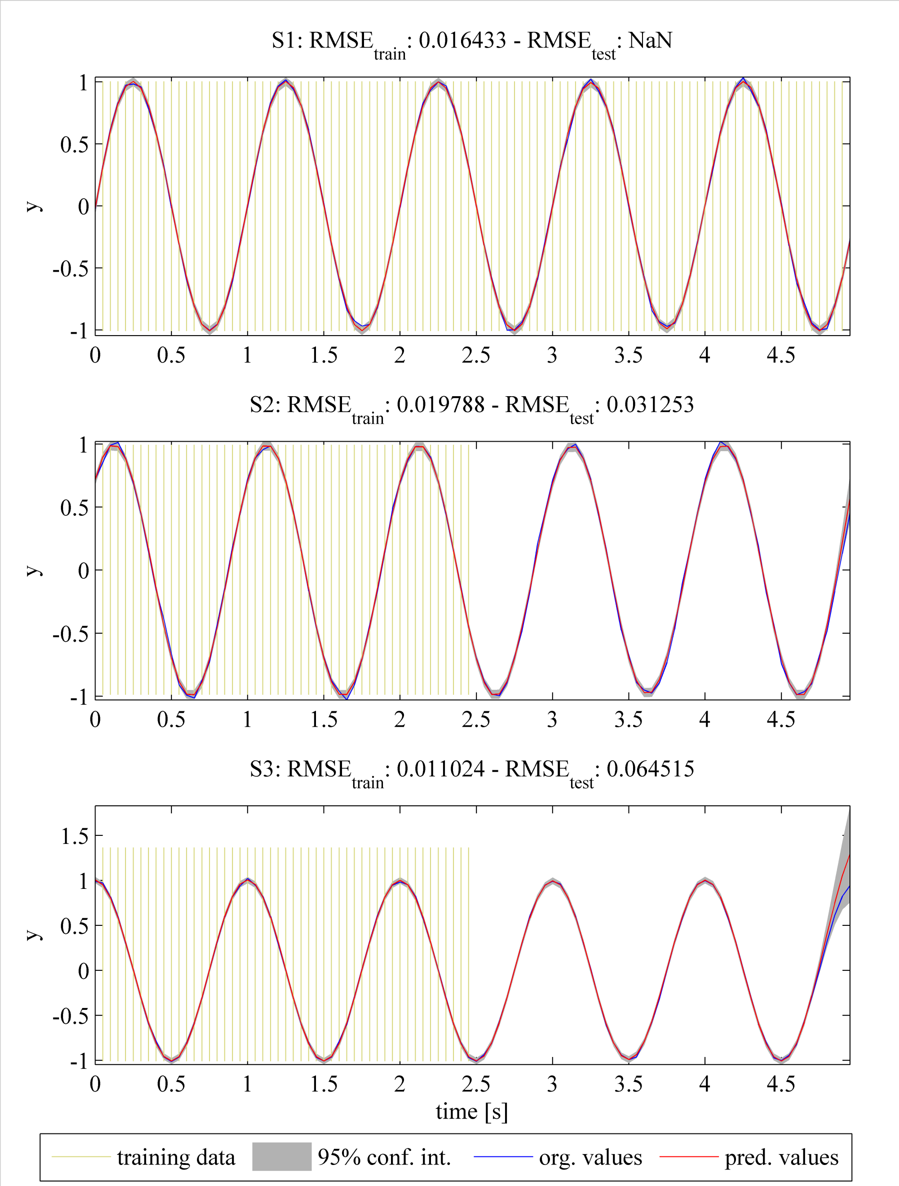
The correlation between tasks can be effected by time-shifts between tasks (e.g., one task is temporally delayed with respect to another). This demo illustrates such an effect (in case 1) and shows how MTGP models can estimate and compensate for these time-shifts between tasks (in case 2). Case 3 then illustrates how this idea could be further used to perform multidimensional template matching. To use *demoMTGP_shift.m*, please include the path of the GPML toolbox or specify the path in *demoMTGP_shift.m* (line 6). The example contains 3 cases. Please select one of the cases by specifying the variable *MTGP_case* (line 21).

line 21: **MTGP_case = 1** % default method



The plots above show the original dataset (shown by the blue solid line) for three sinusoidally-varying tasks (S1 - S3). The tasks are not perfectly correlated with each other because a phase shift has been added to task S2 (phi2 = 45°) and task S3 (phi2 = 90°). The training data for task S1 are taken to be all data occurring within the range of t = [0 5]s. The training data for the other two tasks are taken to be those data between t = [0 2.5]s for both S2 and S3 (marked by the shaded yellow regions). The results have been generated by the MTGP toolbox using a squared exponential temporal covariance function k_t. The model is initialised assuming perfectly correlated tasks. The predicted values are shown by the red solid line and with the 95% confidence interval shown by the shaded grey area. These results indicate that this normal MTGP model (which is not modelling phase shift between tasks) is strongly affected by the phase shifts that exist. This leads to high values of RMSE_test for task S2 and S3, and where it may be seen that the predictions (blue line and grey region) do not closely follow the actual data (red line).

line 21: **MTGP_case = 2**



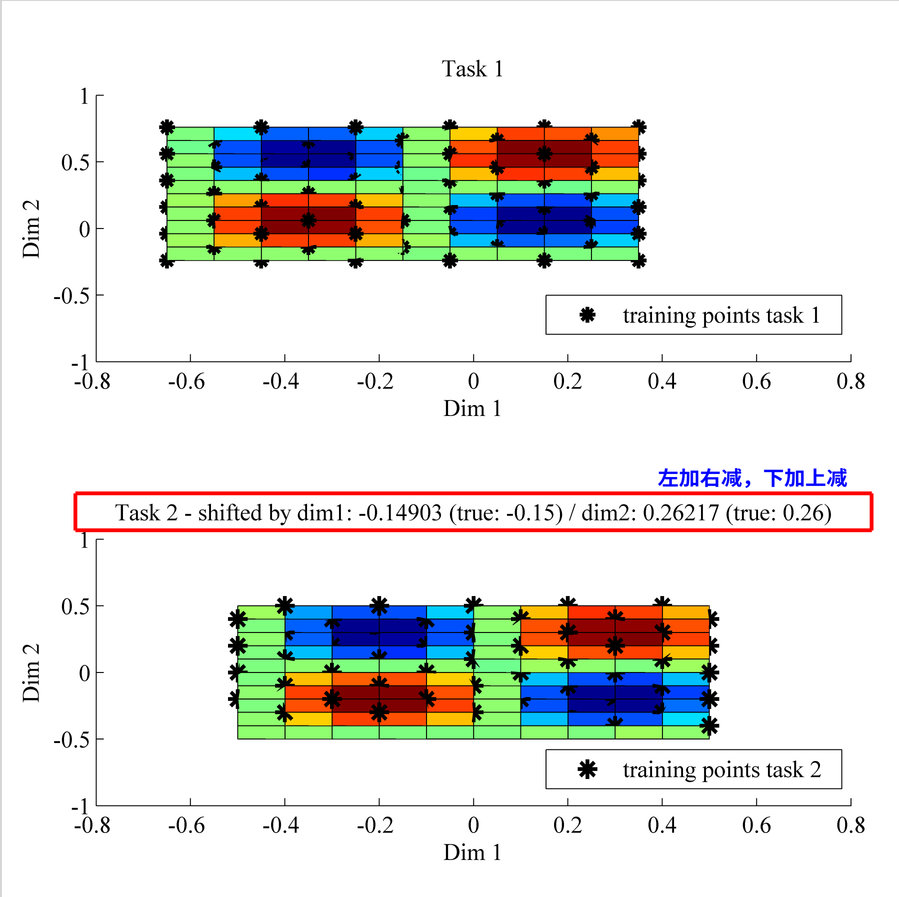
MTGP_covSEisoU_shift

We can improve the accuracy of the predictions shown in case 1 by extending the MTGP with two hyperparameters that represent the phase shift between tasks S1-S2 and tasks S1-S3. The new hyperparameters can be optimised by minimising the NLML as before. As anticipated, we can see that the prediction accuracy increases (with a lower corresponding RMSE_test) and the confidence interval (shown in grey) decreases, representing our increased certainty in the prediction. The optimised phase-shift hyperparameters are shown at the console: theta_s2 = 0.126s and theta_s3 = 0.25s, which is equivalent to the phase shifts phi2 and phi3 for sinusoidal signals with a period of 1s (as we have in this example).

T = 1/f; f = 1Hz; T = 1s
ω = 2πf = 2π rad/s

π/4/ω*T = π/8*1 = 0.125s
π/8/ω*T = π/4*1 = 0.25s

line 21: **MTGP_case = 3**



The above example can easily be extended from univariate time-series to multivariate quantities (e.g., over spatial coordinates). This case illustrates an example how MTGP could be used for template matching with a bivariate data space. We emphasise that, for this example, tasks S1 and S2 do not have any training data in common. The figures show two 2-dimensional tasks. Task S2 is shifted with respect to task S1 in the x-dimension by -0.15 and in the y-dimension by -0.26. The training points for each task are marked as black asterisks. A MTGP model that has two additional hyperparameters (one for each dimension) is used. As shown in the lower sub-plot (within the red box), the estimates for the x- and y-shift between task S1 and S2 are very close to the actual offsets.

Return to MTGP toolbox page...

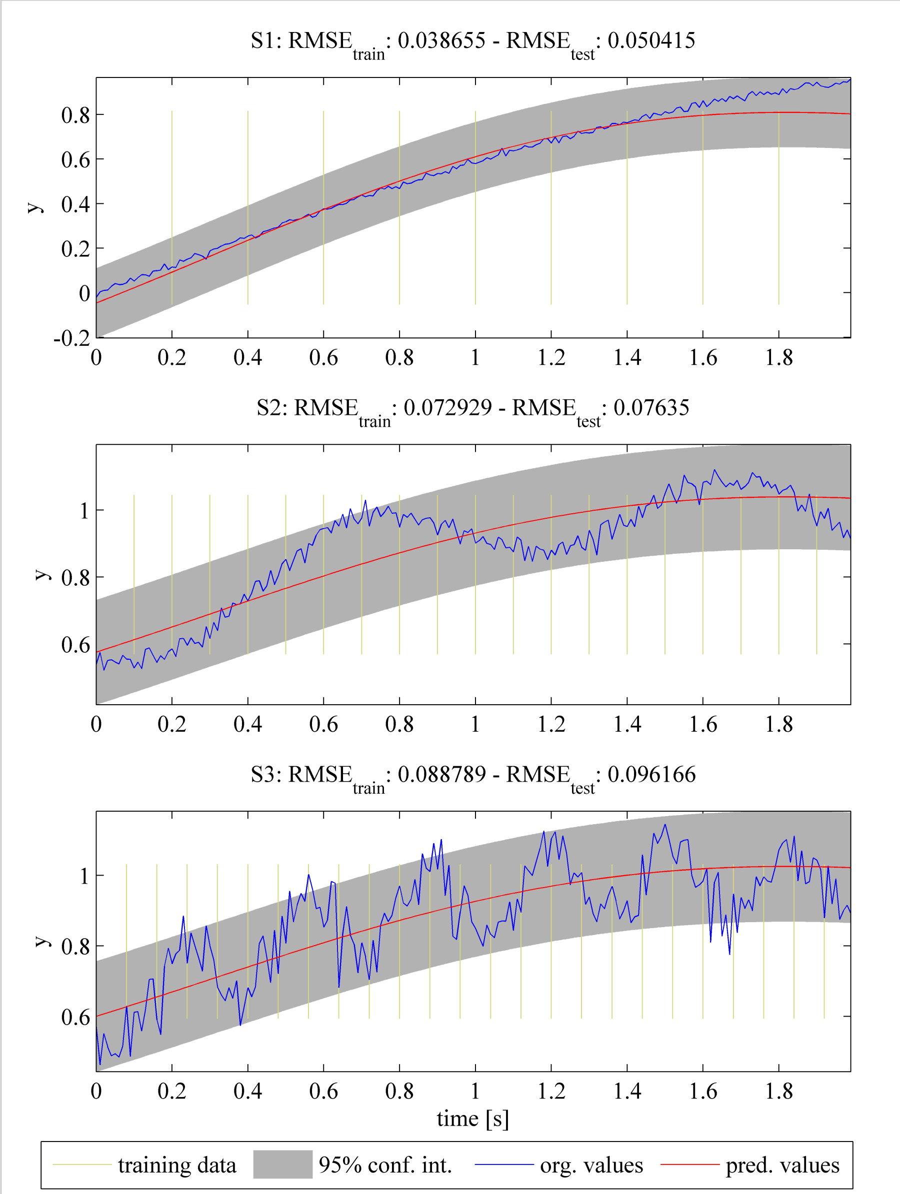
Demo 3

Tasks with different temporal characteristics

File name: *demoMTGP_convoluted_kernels.m*

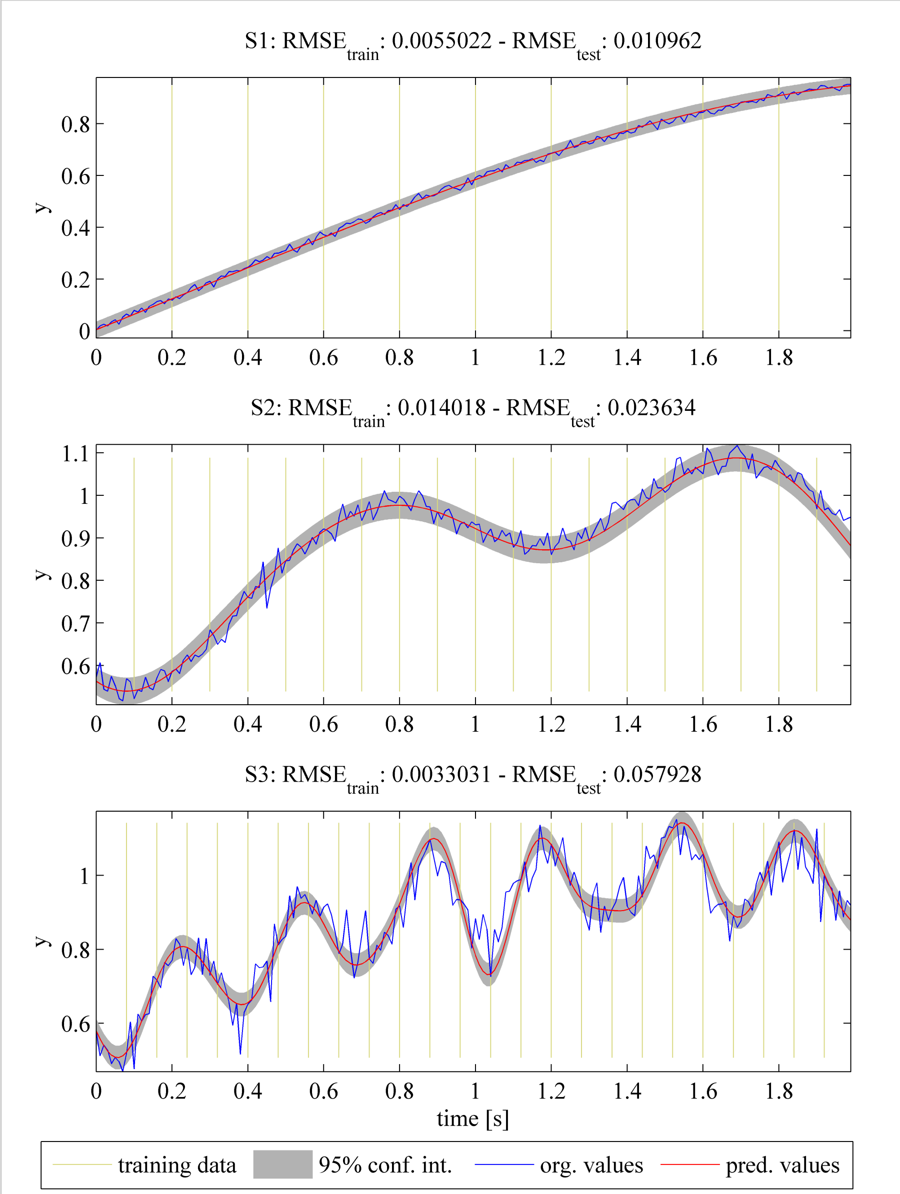
So far, it has been assumed that all tasks share similar temporal characteristics (e.g., that periodicity and noise can be described by the same covariance function for each task). This demo illustrates how the MTGP framework can cope with tasks that have different characteristics (in case 1) and shows how MTGP models can be further modified to allow task-specific temporal characteristics (in case 2). This is achieved using convoluted kernels. To use *demoMTGP_convoluted_kernels.m*, please include the path of the GPML toolbox or specify the path in *demoMTGP_convoluted_kernels.m* (line 9). The example contains 2 cases. Please select one of the cases by specifying the variable *MTGP_case* (line 22).

line 22: **MTGP_case = 1** % default method



The plots above show the original dataset (shown by the blue solid line) of three tasks (S1 - S3). Tasks S2 and S3 were generated by adding an additional sinusoidal component (with a small amplitude and with Gaussian noise) to task S1. Each task is sampled at a different frequency (marked by the vertical yellow lines). The predictions were generated by the MTGP toolbox using a squared exponential temporal covariance function k_t. The model was initialised assuming perfectly-correlated tasks. The predictions are shown by the red solid line and the 95% confidence interval is shown by the shaded grey area. In this case, each task is constrained to share the same values for their respective hyperparameters, and it may be seen that the predictions provided by the the MTGP model are a compromise over all tasks. The smaller variations in the amplitude of task S2 and S3 cannot be modelled, and it may be seen that the dynamics of task S1 are dominating.

line 22: **MTGP_case = 2**



If the variations in the amplitude of tasks S2 and S3 are to be modelled within a single model, the MTGP can be extended by using task-specific hyperparameters. This is possible by using a convolution of covariance functions. The figures show the predictions for the tasks presented in case 1. Now, however, each task has a task-specific covariance hyperparameter. It may be observed that each task can be modelled more precisely. The values of the optimised hyperparameters representing the temporal scaling are 1.45, 0.47, 0.08 for tasks S1, S2, and S3, respectively. As expected, this time-scale parameter decreases as the dynamics of the task become more rapidly-changing.

Return to MTGP toolbox page...

Demo 4

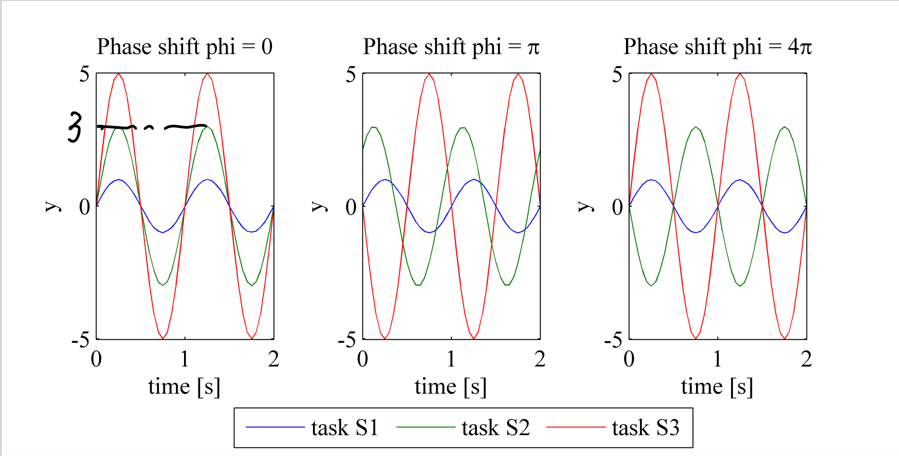
Interpreting the correlation coefficients

File name: *correlation/demoMTGP_normalizedcorrelation_phaseshift.m*

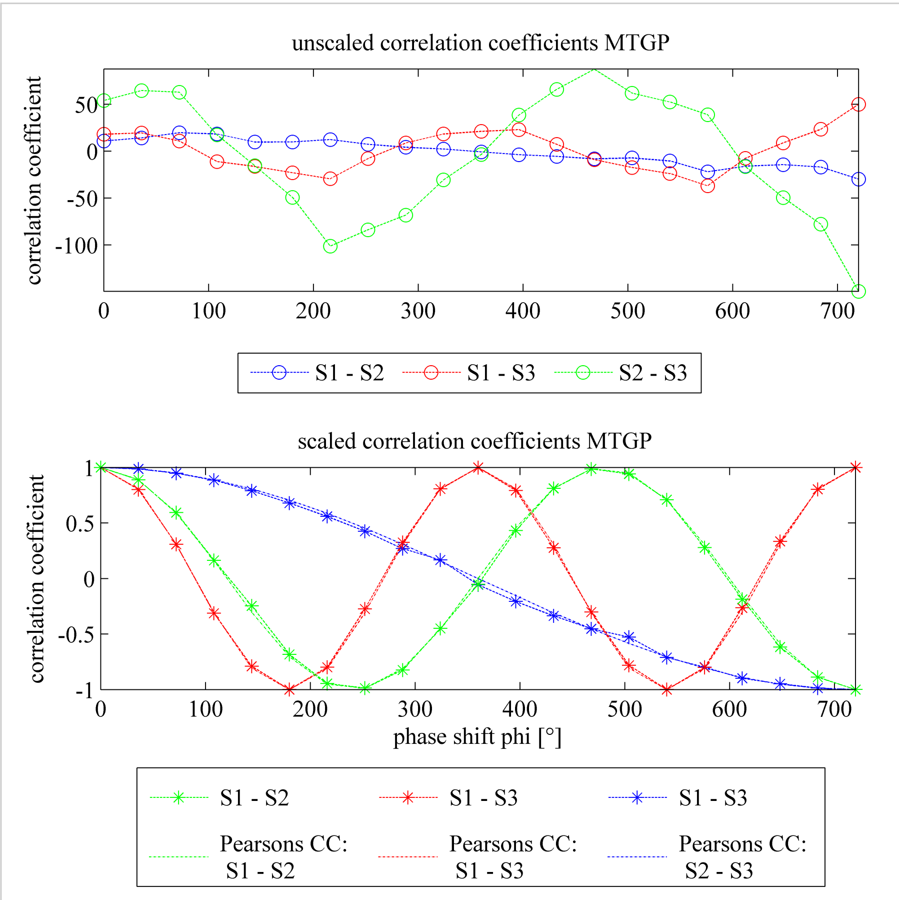
MTGP models are based on the correlation between tasks. Therefore, the analysis of correlation coefficients between tasks should be possible. However, the correlation matrix K_c is effected by the y-scaling of the tasks. This demo contains an example that illustrates how the correlation coefficients can be scaled to lie in the range [-1,1] and which may then be compared to the corresponding value of Pearson's correlation coefficient. To use *demoMTGP_normalizedcorrelation_phaseshift.m*, please include the path of the GPML toolbox or specify the path in *ddemoMTGP_normalizedcorrelation_phaseshift.m* (line 15).

Pearson's correlation coefficient

$$\rho_{X,Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$



In this example, three sinusoidal tasks are given (see plots above). All points are assumed to be available as training data, and the MTGP model is initialised assuming perfectly-correlated tasks. To investigate the correlation between the task, we add a phase shift phi = [0 4pi] to tasks S2 and S3. For each value of phi within that range, task S2 is shifted by phi/4 and task S3 by phi. The plots above show the training data when phi = 0 (left), phi = pi (middle), and phi = 4pi (right). In contrast to demo 2, no ~~temporal shifts~~ are considered here.



The plots above show the computed correlation coefficients between tasks S1-S2 (shown in blue), tasks S1-S3 (shown in red), and tasks S2-S3 (shown in green) of the MTGP model, for each value of phi in the range phi = [0 4pi]. The upper plot shows the unscaled correlation coefficients obtained from the MTGP model. The correlation coefficients are strongly affected by the y-scaling of the three tasks. A direct comparison to the Pearson's correlation coefficient is difficult as the coefficients from the MTGP do not lie within the range [-1,1]. The lower plot shows the MTGP correlation coefficient after normalisation. It may be observed that this now closely follows the actual value of Pearson's correlation coefficient between each pair of tasks.

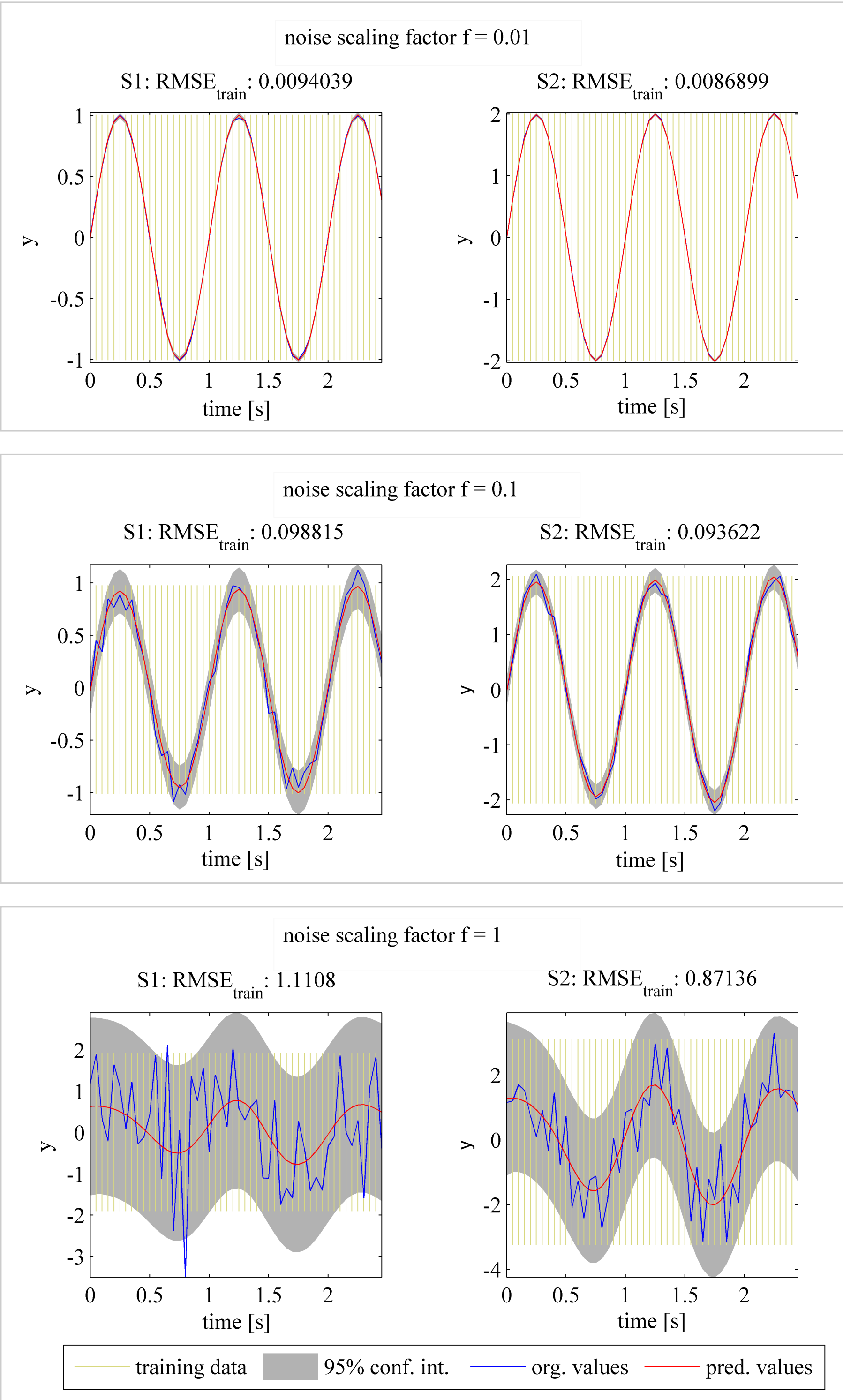
Return to MTGP toolbox page...

Demo 5

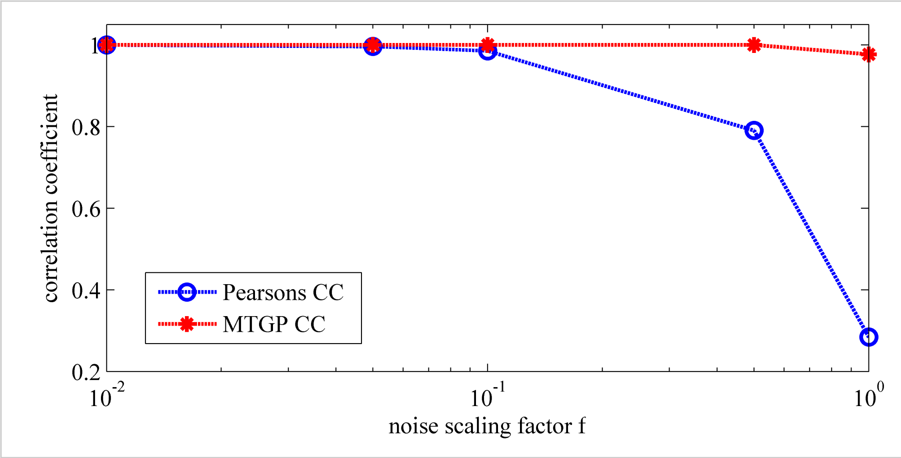
Noise free correlation analysis

File name: *correlation/demoMTGP_noisefree_correlation.m*

Based on the correlation results of demo 4, more complex correlation analysis can be performed. This demo illustrates how MTGPs may be used to analyse the correlation between two tasks independently of the measurement noise. To use *demoMTGP_noisefree_correlation.m*, please include the path of the GPML toolbox or specify the path in *demoMTGP_noisefree_correlation.m* (line 15).



We assume the scenario that two tasks (S1 and S2) are given which were acquired with the same measurement equipment (e.g., two EEG channels measured with the same signal amplifier). The aim is to compute the correlation between these two tasks independently of the noise of the measurement equipment. To simulate this, we assume two sinusoidal signals are given (as shown in the upper plot, above), which are perfectly correlated. We add Gaussian noise N(0,f) to both tasks, where f = [0.01, 0.05, 0.1, 0.5, 1]. The first row of the above shows the original dataset (blue solid line) with f = 0.01; the second row of the above shows the data with f = 0.1, and the third row shows the data with f = 1. For each case, a MTGP model can be trained. The predictions are shown by the red solid lines and the confidence intervals by the grey shaded areas. As the noise on the signals is increased, the confidence intervals on the predictions increases as anticipated. However, it may be seen that the mean prediction results still describe the sinusoidal behaviour of both tasks.



This above plot shows the normalised MTGP correlation coefficient and the Pearson's correlation coefficient between tasks S1 and S2 depending on the noise scaling factor f. It may be observed that with increasing f, the value of Pearson's correlation coefficient decreases due to the increasing noise. In contrast, the normalised MTGP correlation coefficients remains high and seems to be less dependent on the noise factor f, demonstrating increased robustness of the MTGP's estimate of the correlation between tasks.

Return to MTGP toolbox page...

Updated May, 2014