Microevolutionary system identification and climate response predictions by use of BLUP prediction error method

R. Ergon [1]

[1] *Department of Electrical engineering, Information Technology and Cybernetics, University of South-Eastern Norway, Porsgrunn, Norway. E-mail: rolf.ergon@usn.no*

Abstract

Animals and other organisms in wild populations may adjust to climate change by means of plasticity and evolution, and it is an important task to find the contributions from each of these effects. Attempts to solve this disentanglement problem by use of best linear unbiased prediction (BLUP) and restricted maximum likelihood (REML) methods, as borrowed from the field of domestic breeding, have been criticized because of errors in the variances of the predicted random effects. A primary purpose of this article is to show how the problem can be solved by use of BLUP in a prediction error method (PEM), borrowed from the well-established engineering system identification discipline.

The PEM approach is first to collect environmental input data u_t and mean phenotypic output data $\overline{y}_t$, as well as individual phenotypic and fitness data, for consecutive generations from $t = 1$ to T. A reaction norm model of the evolutionary system is then used to find predictions $\widehat{\overline{y}}_t$, and the parameters in this model, together with environmental reference values and initial state variables, are finally tuned such that $\sum_{t=1}^{T} \left(\overline{y}_t - \widehat{\overline{y}}_t \right)^2$ is minimized.

The main contribution is the use a dynamical BLUP model in a BLUP/PEM method for parameter estimation and mean reaction norm trait predictions. The model is dynamical in the sense that the incidence matrix in an underlying linear mixed model, as well as the corresponding residual covariance matrix, are functions of time. For comparisons, a selection gradient prediction model as presented in Ergon (2022a,b) is also used in a GRAD/PEM method. The advantages of the BLUP/PEM method are that it can utilize genetic relationship information, and that it produces better estimates of environmental reference values. The treatment is limited to multiple-input single-output (MISO) systems. Generations are assumed to be non-overlapping.

Simulation examples show that BLUP/PEM may find good estimates of environmental reference values and initial state variables, as well as good mean reaction norm trait predictions. Details for use of additional fixed effects, as well as appropriate methods for model validation remain to be worked out.

Keywords: Reaction norm evolution, dynamical BLUP model, system identification, prediction error method

---

1. Introduction

Individuals in a population of wild organisms may adjust to climate change by means of phenotypic plastic-ity, i.e., by changes in individual behavioral, morphological and physiological traits in response to changes in the environment, without natural selection being involved. A population of wild organisms may also

adapt to climate change by evolution, i.e., by natural selection such that the proportion of individuals with favourable traits increases from generation to generation. The combined effect of plasticity and evolution can, for example, be seen in that many bird species have advanced their breeding period by several weeks in the last decades (see, e.g., Bowers et al., 2016). It is well documented that individual phenotypic plasticity plays an important role in such acclimations (Merilä and Hendry, 2014), but in many cases it is unclear to which extent evolution plays a role, and this is further complicated by the fact that also plasticity may evolve. Methods for disentanglement of plasticity and evolutionary effects on acclimation to climate change is therefore an active research area (see, e.g., Scheiner et al., 2020). A main purpose of this article is to show that system identification methods from the field of engineering control can play an important role in such disentanglement efforts.

The combined effect of plasticity and evolution is often modeled by use of reaction norms, for example the two-trait intercept-slope model as used in Lande (2009),

$$y_{i,t} = a_{i,t} + v_{i,t} + (b_{i,t} + \eta_{i,t})(u_t - u_{\text{ref}}), \qquad (1a)$$

where $y_{i,t}$ is the individual phenotypic response on an environmental cue $u_t - u_{\text{ref}}$ at generation t (where Lande assumed $u_{\text{ref}} = 0$ and $\eta_{i,t} = 0$), and where u_{ref} is a reference environment as discussed below. Note that I in the following will treat the reaction norm parameters $a_{i,t} + v_{i,t}$ and $b_{i,t} + \eta_{i,t}$ as quantitative traits in their own right, and where necessary I will thus distinguish between phenotypic traits and reaction norm traits. In Equation (1a), $a_{i,t}$ and $b_{i,t}$ are the additive genetic components of the reaction norm traits, while $v_{i,t}$ and $\eta_{i,t}$ are independent and identically distributed (iid) zero mean normal non-additive effects (genetic or environmental white noise). From Equation (1a) follows the mean trait equation

$$\overline{y}_t = \overline{a}_t + \overline{b}_t(u_t - u_{\text{ref}}), \qquad (1b)$$

where the mean traits $\overline{a}_t$ and $\overline{b}_t$ may evolve as results of natural selection (Lande, 2009).

The plasticity-evolution disentanglement problem for a population system given by Equations (1a,b) is illustrated in Figure 1, where an individual Gaussian fitness function is introduced. Fitness is here a measure of reproductive success (Ergon, 2019, 2022a), for example number of offspring. Initially, it is assumed that the population is fully adapted to the temperature $u_{\text{ref}} = 10$ ℃, in the sense that the expected geometric mean fitness has a global maximum at this temperature, which implies that the variance of $y_{i,t}$ has a global minimum (Lande, 2009). Panel b) shows

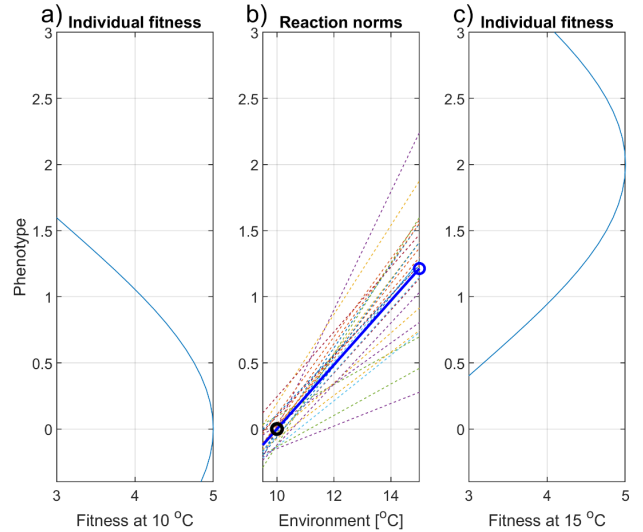Figure 1: Illustration of phenotypic and fitness changes as results of a sudden change in mean environment from 10 ℃ to 15 ℃. Panels a) and c) show the individual fitness functions at 10 ℃ and 15 ℃, respectively. Panel b) shows individual and mean reaction norms in a stationary stochastic environment with the expected temperature $\text{E}[u_t] = u_{\text{ref}} = 10$ ℃ (dotted lines and solid line, respectively), from which follows that the immediate plastic response on a mean temperature change from 10 ℃ to 15 ℃, is a change in expected mean phenotypic value from 0 to 1.2 (circles).

25 realizations of individual reaction norms, as well as the mean reaction norm $\overline{y}_t$, in a stationary stochastic environment with the expected temperature 10 ℃, and thus the expected cue value $\text{E}[u_t - u_{\text{ref}}] = 0$. As shown in panel a), this means that $\text{E}[\overline{y}_t] = 0$ is equal to the expected phenotypic value that maximizes fitness, such that the variance in individual fitness values is at a minimum. If the temperature then suddenly changes to 15 ℃, the expected mean phenotypic value will immediately change to $\text{E}[\overline{y}_t] = 1.2$, and this will be a purely plastic response. If at the same time the peak of the fitness function is assumed to move from 0 to 2, as shown in panel c), it follows that the mean fitness can be improved by evolution of both $\overline{a}_t$ and $\overline{b}_t$, such that $\text{E}[\overline{y}_t]$ finally reaches the value 2. Note that the mean fitness will then be lower than the initial mean fitness, because the variance in individual fitness values will be larger. Concomitant with a settlement on $\text{E}[\overline{y}_t] = 2$, the value of $\text{E}[\overline{b}_t]$ will return to the initial optimal value in a stationary stochastic environment (Lande, 2009). In the end, 100% of the change in $\text{E}[\overline{y}_t]$ will thus be due to evolution of $\overline{a}_t$, while 0% is

due to evolution of $\bar{b}_t$, but evolution of $\bar{b}_t$ may still play a vital role in the period before settlement in the new environment. Evolution of $\bar{b}_t$ may in fact be crucial for survival of the species. See also Lande (2009) for simulations of the mean reaction norm trait responses in a case as described. In the example as illustrated in Figure 1, there is the possibility that the temperature u_{ref} which the population is fully adapted to also is an evolvable trait, in addition to the mean intercept and mean plasticity slope. Such cases are discussed in Ergon and Ergon (2017) and Ergon (2018).

The first essential question in this article is how the evolution of the mean reaction norm traits $\bar{a}_t$ and $\bar{b}_t$ in a population can be found from recorded values of u_t, $y_{i,t}$, and individual fitness, and then also in more realistic cases with a gradual change in a multivariate environment. The second essential question is how a model for this purpose can be identified in a way such that genetic relationships between the individuals in the population are taken into account. As discussed below, also the reference environment and initial mean reaction norm trait values may be identified from time series data.

The adaptive contribution from evolution can be predicted by use of a linear mixed model with fixed and random effects, combined with BLUP (best linear unbiased predictions) and REML (restricted maximum likelihood) methods, as extensively used in domestic breeding (Henderson, 1950; Ch. 26 and 27 Lynch and Walsh, 1998). BLUP/REML applications on wild animal populations are for example discussed in Kruuk (2004) and Nussey et al. (2007), while Arnold et al. (2019) discusses plant applications. Hadfield et al. (2010) criticized this approach on the ground that BLUP underestimates the variances of the random effects in the linear mixed model. However, as shown in Ergon (2022c), this underestimation is precisely what is needed, in order to obtain the correct incremental changes for prediction of phenotypic traits of non-plastic organisms, and as shown in Ergon (2023), this is the case also for reaction norm traits of plastic organisms.

With known u_t and $\bar{y}_t$ time series data, the traditional BLUP/REML breeding problem is turned into what in the engineering control community is called a system identification problem. As will be shown in Section 2, the use of input data u_t in a linear mixed model leads to a time varying coincidence matrix $\widetilde{Z}_t$, and REML at a single generation cannot therefore be used for parameter estimation, as it is otherwise done for BLUP parameter estimation. An alternative is therefore to apply REML on data from several generations, or to use a prediction error method (PEM) that utilizes data from all generations in a study.

System identification can be described as the *art* and *science* of building mathematical models of dynamical systems from observed input-output signals, and it has a wide range of applications in various fields of physics and engineering (Ljung, 2010; Ljung et al., 2011). PEM is one of the important methods in system identification (Ch. 7-9, Ljung, 1999; Ch. 7, Stoica and Söderström, 1989), and a primary purpose of this article is to show that BLUP/PEM is a feasible solution to the problem with a time varying design matrix in a linear mixed model. Note that both REML and PEM make use of various versions of the numerical Newton algorithm, and that there are clear theoretical links between ML and PEM (Ljung, 2010; Ch. 7, Stoica and Söderström, 1989).

I will thus show how the evolutionary system identification problem can be solved by use of BLUP models in PEM. The treatment will be limited to the relatively simple cases where a population evolves in response to changes in a physical multivariate environment (microevolution). The more realistic cases with evolutionary interaction between several species (macroevolution) may be far more involved. I will discuss evolutionary systems with multiple environmental input signals and a single mean phenotypic output signal (MISO systems), i.e., extensions of the model in Equations (1a,b). I will thus assume an individual multivariate intercept-slope reaction norm model

$$y_{i,t} = a_{i,t} + v_{i,t} + (b_{1,i,t} + \eta_{1,i,t})\, u'_{1,t}$$
$$+ (b_{2,i,t} + \eta_{2,i,t})\, u'_{2,t} + \cdots + (b_{q,i,t} + \eta_{q,i,t})\, u'_{q,t}, \tag{2a}$$

where $y_{i,t}$ is the individual phenotypic response on environmental cues $u'_{1,t} = u_{1,t} - u_{1,ref}$, $u'_{2,t} = u_{2,t} - u_{2,ref}$, $\ldots$, $u'_{q,t} = u_{q,t} - u_{q,ref}$, where $u_{1,ref}$, $u_{2,ref}$, $\ldots$, $u_{q,ref}$ define the reference environment. Note that such reference values are needed for all interval scaled input variables, as for example temperature in ℃, and that they must be properly defined (Ergon, 2022a). Ratio scaled input variables have a well-defined zero-point, but it follows from the discussion in Ergon (2022a) that these zero-points are not necessarily the proper reference values. With multiple input variables, the reference environment is defined by a column vector $\boldsymbol{u}_{\mathrm{ref}}$.

In Equation (2a), $a_{i,t}$, $b_{1,i,t}$, $b_{2,i,t}$, $\ldots$, $b_{q,i,t}$ are the additive genetic components of normally distributed individual reaction norm intercept and plasticity slopes, respectively, and it is thus the mean values $\bar{a}_t$, $\bar{b}_{1,t}$, $\bar{b}_{2,t}$, $\ldots$, $\bar{b}_{q,t}$ that are evolvable. The variables $a_{i,t}$, $b_{1,i,t}$, $b_{2,i,t}$, $\ldots$, $b_{q,i,t}$ may be correlated, as determined by an additive genetic covariance matrix $\boldsymbol{G}$. The terms $v_{i,t}$, $\eta_{1,i,t}$, $\eta_{2,i,t}$, $\ldots$, $\eta_{q,i,t}$ are iid zero mean normal non-additive effects (genetic and environmental white

85

noise). As done in Lande (2009) and Ergon and Ergon (2017), the variables $a_{i,t}+v_{i,t}$, $b_{1,i,t}+\eta_{1,i,t}$, $b_{2,i,t}+\eta_{2,i,t}$, ..., $b_{q,i,t} + \eta_{q,i,t}$ are treated as quantitative traits in their own right. These traits have a covariance matrix $\boldsymbol{P} = \boldsymbol{G} + \boldsymbol{H}$, where $\boldsymbol{H}$ is a diagonal matrix with variance values σ_v^2, $\sigma_{\eta_1}^2$, $\sigma_{\eta_2}^2$, ..., $\sigma_{\eta_q}^2$ along the main diagonal. The time unit is generation, and generations are assumed to be non-overlapping.

From Equation (2a) follows the equation for the mean phenotypic value,

$$\overline{y}_t = \overline{a}_t + \begin{bmatrix} u'_{1,t} & u'_{2,t} & \cdots & u'_{q,t} \end{bmatrix} \begin{bmatrix} \overline{b}_{1,t} \\ \overline{b}_{2,t} \\ \vdots \\ \overline{b}_{q,t} \end{bmatrix}$$

$$= \overline{a}_t + (\boldsymbol{u}_t - \boldsymbol{u}_{\text{ref}})^T \overline{\boldsymbol{b}}_t. \qquad (2b)$$

Here, the mean reaction norm traits may be collected in a vector $\overline{\boldsymbol{x}}_t = \begin{bmatrix} \overline{a}_t & \overline{\boldsymbol{b}}_t^T \end{bmatrix}^T$. The theory will be developed under the assumption of constant matrices $\boldsymbol{G}$ and $\boldsymbol{P}$, but a case with time-varying matrices will be tested in simulations.

Although the main purpose is to show how a BLUP model can be incorporated into the PEM approach, I will also compare with results found by a selection gradient (GRAD) method as introduced in Ergon (2022a), and as further developed in Ergon (2022b). This method is based on the multivariate breeder's equation (Lande, 1979). Principal differences between the two approaches will be further discussed below.

Independent of the details of the PEM identification method, there are some general system identification problems that must be considered. First, the initial state of the system affects the dynamics, and this state must therefore be known or identified (Liu, 2013). A complete identification of an unknown dynamical reaction norm system will thus require that also the initial mean reaction norm traits are estimated.

Second, PEM uses a mathematical model with the same input signal as the true system, such that the system can be identified by parameter optimization (model tuning) until the model output in some sense is as close as possible to the true output. For reaction norm models this implies that also the environmental reference vector $\boldsymbol{u}_{\text{ref}}$ must be known or estimated, and that mean centering of for example a temperature trend is not a good idea (Ergon, 2022a). Such mean centering appears to be common in applications of linear mixed models in evolutionary contexts, but that assumes stationary stochastic environments (e.g., Thomson et al., 2018). In practice it may be difficult to find the reference point in the environment space, and it is even more difficult because it may evolve (Er-

gon and Ergon, 2017; Ergon, 2018). I will, however, assume a constant reference environment.

Third, the aspect of system identifiability is essential, and an identifiability analysis of the GRAD and BLUP microevolutionary models is therefore a part of Section 2, including identifiability of the reference environment and initial mean reaction norm trait values. Here, note that traditional REML as used in domestic breeding (Ch. 27, Lynch and Walsh, 1998) would have to be modified for estimation of the reference environment. As verified by simulations, such an estimation is possible in the BLUP/PEM approach.

As we will see, microevolutionary PEM relies on measurements of the environmental input variables, as well as on individual phenotypic values and fitness, and modeling errors and noise corrupted measurements can make the estimation of especially $\boldsymbol{u}_{\text{ref}}$ difficult. As explained in Section 2, and shown in simulations, this is a serious problem in the GRAD/PEM approach, while the BLUP/PEM approach is much less sensitive. A main question is still how errors in $\boldsymbol{u}_{\text{ref}}$ affect the mean reaction norm trait predictions.

An additional aspect is that although dynamical systems to some extent can be identified from output-only measurements, it will in practice often be impossible to find good parameter estimates unless the system is persistently excited by the input signals (Ch. 13, Ljung, 1999). This is hardly a problem when we as in the simulations in Section 3 assume yearly temperatures in for example wild bird habitats as part of the environmental input signals, and persistent excitation of sufficient order will therefore be assumed.

The PEM principle is simple, as illustrated in Figure 2. For readers who are not accustomed to block diagrams, it should be pointed out that the block with an adjustable dynamical system model (the tuning model) is just an illustration of either the GRAD or dynamical BLUP prediction equations, as developed in Section 2. These equations make use of environmental input data $\boldsymbol{u}_t$, and phenotypic output data $\boldsymbol{y}_t$ (a vector of individual values) and $\overline{y}_t$ (the population mean value) from the microevolutionary system, which are collected for a number T of consecutive generations. In addition, a vector $\boldsymbol{w}_t$ of individual relative fitness values must be collected for each generation, where the individual fitness is a measure of reproductive success. The relative fitness is simply the individual fitness divided by the mean fitness in the population under study.

Note that the individual fitness values are used as inputs to the tuning model, and that the tuning model therefore does not include a fitness function. In simulations, however, an individual fitness function is needed in the evolutionary model that generates true data. As illustrated in Figure 1., this function must define the

phenotypic value that gives maximum fitness as function of the environment.

The inputs $\boldsymbol{u}_t$, $\boldsymbol{y}_t$ and $\boldsymbol{w}_t$ to the tuning model (the GRAD or BLUP prediction equations) result in the predicted mean phenotype $\widehat{\overline{y}}_t$, such that the prediction error $\overline{y}_t - \widehat{\overline{y}}_t$ can be computed. In BLUP/PEM, also a known additive genetic relationship matrix $\boldsymbol{A}_t$ is necessary (Ch. 26, Lynch and Walsh, 1998). The model parameters, including $\boldsymbol{u}_{\mathrm{ref}}$ and the initial mean reaction norm traits, are finally adjusted such that $\sum_{t=1}^{T} \varepsilon_t^2 = \sum_{t=1}^{T} \left(\overline{y}_t - \widehat{\overline{y}}_t \right)^2$ is minimized. The *art* part of this solution is to formulate a tuning model that does not have unnecessarily many parameters to estimate, while it at the same time is rich enough to capture the essential properties of the microevolutionary system.
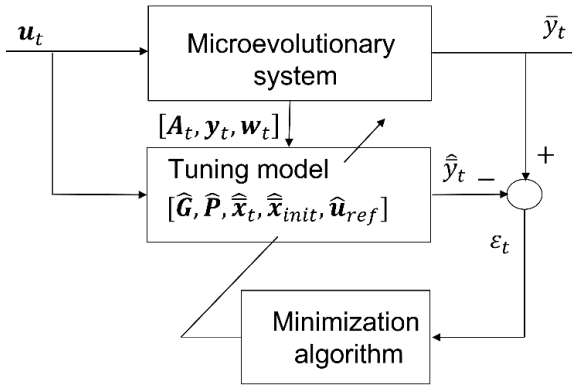

Figure 2: Block diagram of microevolutionary GRAD/PEM and BLUP/PEM for a MISO system, with dynamical tuning model based on a reaction norm model with mean traits vector $\overline{\boldsymbol{x}}_t = \begin{bmatrix} \overline{a}_t & \overline{b}_{1,t} \cdots & \overline{b}_{q,t} \end{bmatrix}^T$. Here, $\boldsymbol{u}_t$ and $\overline{y}_t$ are the known environmental input vector and the known mean phenotypic value at time t, respectively. $\boldsymbol{A}_t$ is the known additive genetic relationship matrix, while $\boldsymbol{y}_t$ and $\boldsymbol{w}_t$ are known vectors of individual phenotypic and relative fitness values, respectively. The $\widehat{\boldsymbol{G}}$ and $\widehat{\boldsymbol{P}}$ matrices include the system parameters, while $\widehat{\overline{\boldsymbol{x}}}_{\mathrm{init}}$ and $\widehat{\boldsymbol{u}}_{\mathrm{ref}}$ are vectors defining the initial mean reaction norm trait values and the reference environment, respectively. All of these model parameters are tuned until $\sum_{t=1}^{T} \varepsilon_t^2 = \sum_{t=1}^{T} \left(\overline{y}_t - \widehat{\overline{y}}_t \right)^2$ is minimized.

The microevolutionary prediction error method as illustrated in Figure 2 will, as already mentioned, be investigated with use of two types of tuning models, where predictions $\widehat{\overline{y}}_t = \widehat{\overline{a}}_t + (\boldsymbol{u}_t - \widehat{\boldsymbol{u}}_{\mathrm{ref}})^T \widehat{\overline{\boldsymbol{b}}}_t$ are found in

different ways. Primarily, $\widehat{\overline{y}}_t$ is found from a dynamical BLUP model, which makes use of genetic relationship information. For comparisons, $\widehat{\overline{y}}_t$ is also found by use of a selection gradient (GRAD) model, as developed for a SISO system in Ergon (2022a), which ignores information on genetic relationships in the population.

After tuning of the prediction model in Figure 2, i.e., when $\widehat{\overline{y}}_t \approx \overline{y}_t$ for all generations from $t = 1$ to T, predictions of the mean reaction norm traits can be recovered from the model. The quality of these predictions with random measurement errors in the $\boldsymbol{u}_t$ and $\boldsymbol{y}_t$ data will here be studied by means of simulations (while $\boldsymbol{w}_t$ will be the relative number of offspring, which will be assumed to be correctly counted). Prediction errors caused by time-varying parameter values and modeling errors will also be studied.

Section 2 defines the covariance structure for Equation (2a), and gives the theoretical background for the alternative GRAD and BLUP optimization algorithms used for tuning according to Figure 2. A theorem shows that the element $G_{aa} = \mathrm{E}\left[(a_{i,t} - \overline{a}_t)^2 \right]$ in the additive genetic covariance matrix $\boldsymbol{G}$ may be set to any value, such that the rest of the parameters in $\boldsymbol{G}$ and $\boldsymbol{P}$ will be estimated in relation to that value. A second theorem shows that a genetic relationship matrix $\boldsymbol{A}_t = \boldsymbol{I}_N$ results in identical GRAD and BLUP mean reaction norm trait predictions. A third theorem shows that the GRAD prediction model, as well as a BLUP prediction model with $\boldsymbol{A}_t = \boldsymbol{I}_N$ are identifiable, and as supported by simulations it is conjectured that also a BLUP prediction model with $\boldsymbol{A}_t \neq \boldsymbol{I}_N$ is identifiable.

Simulations in Section 3 show the feasibility of GRAD/PEM and BLUP/PEM in the microevolutionary context. The environmental inputs are noisy ramp functions, similar to temperature trends caused by climate change. True mean phenotypic responses $\overline{y}_t$, and true mean trait values $\overline{a}_t$ and $\overline{b}_{1,t}$, $\overline{b}_{2,t}$, ..., $\overline{b}_{q,t}$ are generated by use of the BLUP prediction equations, with known parameter values (including initial mean trait values, reference environment, and additive genetic relationship matrix). Individual additive genetic and environmental effects are at each generation drawn at random from populations with the given covariance matrices $\boldsymbol{G}$ and $\boldsymbol{P}$, modified by the additive genetic relationship matrix $\boldsymbol{A}_t$. With a known model structure, perfect measurements of the input environment, individual fitness, and individual phenotypic values, will result in close to perfect BLUP parameter estimates and predictions $\widehat{\overline{y}}_t$. This makes it possible to study the effects of modeling and measurement errors, and of time-varying parameter values. It also makes it possible to study how ignorance of genetic relationship information affects the parameter estimates and mean reaction norm trait predictions. The population size

in the main simulations is $N = 100$.

A final summary with discussion is given in Section 4. Proof of a theorem is given in Appendix A, while Appendix B shows GRAD results with population size $N = 10,000$. MATLAB simulation code is finally given in Appendix C.

2. Theory

2.1. Covariance structures

For the GRAD and BLUP modeling it is necessary to define the covariance matrices $\boldsymbol{G}$ and $\boldsymbol{P} = \boldsymbol{G} + \boldsymbol{H}$ to be estimated according to Figure 2. For the BLUP modeling it is in addition necessary to introduce Kronecker product covariance matrices $\widetilde{\boldsymbol{G}}_t$ and $\widetilde{\boldsymbol{R}}_t$.

First, the reaction norm traits in Equation (2a) have an $(1 + q) \times (1 + q)$ additive genetic covariance matrix given by the expectation $\boldsymbol{G} = \mathrm{E}\left[[\bullet]^T [\bullet]\right]$, where $[\bullet] = \left(a_{i,t} - \overline{a}_t \quad b_{1,i,t} - \overline{b}_{1,t} \quad b_{2,i,t} - \overline{b}_{2,t} \quad \cdots \quad b_{q,i,t} - \overline{b}_{q,t}\right)$, with block elements $\boldsymbol{G}_{aa}$, $\boldsymbol{G}_{ab}$, $\boldsymbol{G}_{ab}^T$ and $\boldsymbol{G}_{bb}$. The reaction norm traits also have the phenotypic covariance matrix $\boldsymbol{P} = \boldsymbol{G} + \boldsymbol{H}$, where $\boldsymbol{H}$ is a diagonal matrix with variances σ_v^2, $\sigma_{\eta_1}^2$, $\sigma_{\eta_2}^2$, ..., $\sigma_{\eta_q}^2$ along the main diagonal.

Second, a population of N genetically related individuals has the $N(1 + q) \times N(1 + q)$ Kronecker product covariance matrix $\widetilde{\boldsymbol{G}}_t = \boldsymbol{G} \bigotimes \boldsymbol{A}_t$, where $\boldsymbol{A}_t$ is the symmetric $N \times N$ additive genetic relationship matrix (Ch. 26, Lynch and Walsh, 1998; Cantet et al., 2022; Mathew et al., 2018), and where $\bigotimes$ is the Kronecker product operator. The Kronecker product simply means that each of the elements in $\boldsymbol{G}$ are multiplied by $\boldsymbol{A}_t$. If the individuals are genetically unrelated, the additive genetic relationship matrix is a unity matrix, $\boldsymbol{A}_t = \boldsymbol{I}_N$. In BLUP modeling, a population of N individuals also has a residual Kronecker product covariance matrix $\widetilde{\boldsymbol{R}}_t = \boldsymbol{R}_t \bigotimes \boldsymbol{I}_N$, where $\boldsymbol{R}_t$ is the covariance matrix of the residuals in a linear mixed model.

2.2. The GRAD prediction model

The GRAD prediction model is based on the multivariate breeder's equation (Lande, 1979), which means that it implicitly assumes that the additive genetic relationship matrix is a unity matrix (Ergon, 2023). Applied on the reaction norm traits in Equation (2a), the multivariate breeder's equation for the incremental changes

in mean traits from generation to generation becomes

$$\begin{bmatrix} \Delta \overline{a}_t \\ \Delta \overline{b}_{1,t} \\ \vdots \\ \Delta \overline{b}_{q,t} \end{bmatrix} = \boldsymbol{G} \boldsymbol{P}^{-1} \begin{bmatrix} \mathrm{cov}\,(w_{i,t}, a_{i,t}) \\ \mathrm{cov}\,(w_{i,t}, b_{1,i,t}) \\ \vdots \\ \mathrm{cov}\,(w_{i,t}, b_{q,i,t}) \end{bmatrix} \qquad (3)$$

where $w_{i,t}$ i th relative individual fitness, i.e., $w_{i,t} = W_{i,t}/\overline{W}_t$. Note, however, that Equation (3) cannot be used in the tuning model in Figure 2, because the individual traits $a_{i,t}$, etc., are not available. In Ergon (2022a), this problem was solved by use of a linear transformation of the vector $\begin{bmatrix} a_{i,t} & b_{1,i,t} & b_{2,i,t} & \cdots & b_{q,i,t} \end{bmatrix}^T$ onto the vector $\begin{bmatrix} a_{i,t} & b_{1,i,t} & b_{2,i,t} & \cdots & b_{q,i,t} & y_{i,t} \end{bmatrix}^T$. Using that Equation (3) gives

$$\Delta \overline{y}_t = \Delta \overline{a}_t + \boldsymbol{u}'^T_{t+1} \Delta \overline{\boldsymbol{b}}_t + \Delta \boldsymbol{u}'^T_t \overline{\boldsymbol{b}}_t, \qquad (4)$$

where $\Delta \boldsymbol{u}'_t = \boldsymbol{u}'_{t+1} - \boldsymbol{u}'_t$, this leads to

$$\begin{bmatrix} \Delta \overline{a}_t \\ \Delta \overline{b}_{1,t} \\ \vdots \\ \Delta \overline{b}_{q,t} \end{bmatrix} = \begin{bmatrix} \boldsymbol{G}_{aa} & \boldsymbol{G}_{ab} \\ \boldsymbol{G}_{ab}^T & \boldsymbol{G}_{bb} \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{u}'_t \end{bmatrix} \frac{1}{P_{yy,t}} \mathrm{cov}\,(w_{i,t}, y_{i,t}), \qquad (5a)$$

where

$$\boldsymbol{u}'_t = \begin{bmatrix} u_{1,t} - u_{1,\mathrm{ref}} & u_{2,t} - u_{2,\mathrm{ref}} & \cdots & u_{q,t} - u_{q,\mathrm{ref}} \end{bmatrix}^T.$$

Here,

$$P_{yy,t} = G_{aa} + 2\boldsymbol{G}_{ab}\boldsymbol{u}'_t + \boldsymbol{u}'^T_t \boldsymbol{G}_{bb}\boldsymbol{u}'_t + r_t, \qquad (5b)$$

with

$$r_t = \sigma_v^2 + u'^2_{1,t}\sigma_{\eta_1}^2 + u'^2_{2,t}\sigma_{\eta_2}^2 \cdots + u'^2_{q,t}\sigma_{\eta_q}^2. \qquad (5c)$$

Note that Equation (5a) is a dynamical model in the sense that $P_{yy,t}$ varies with time, and that we can set G_{aa} to any fixed value and scale the rest of the parameters accordingly. Also note that Equations (3) and (5a) give identical results only asymptotically, i.e., for $N \to \infty$. In order for Equation (5a) to be compatible with the Price equation (Price et al., 1970; Ergon, 2019), the covariance expression should be computed as

$$\mathrm{cov}\,(w_{i,t}, y_{i,t}) = \frac{1}{N} \sum_{i=1}^{N} w_{i,t} y_{i,t} - \overline{w}_t \overline{y}_t. \qquad (5d)$$

With given initial mean trait values, Equation (5a) gives $\overline{a}_t$, $\overline{b}_{1,t}$, etc., and thus also $\overline{y}_t = \overline{a}_t + (\boldsymbol{u}_t - \boldsymbol{u}_{\mathrm{ref}})^T \overline{\boldsymbol{b}}_t$ according to Equation (2b), for use in

the tuning model in Figure 2. Initial values of the elements in $\bar{\boldsymbol{b}}_t$ for $t = 1$ must be found as part of the identification, while $\bar{a}_1$ follows from Equation (2b) by setting $\bar{y}_1 = 0$. See Section 3 for a simulation example, where GRAD estimates of $\boldsymbol{u}_{\text{ref}}$, $\bar{\boldsymbol{b}}_1$, and the parameters in $\boldsymbol{G}$ and r_t, are found by use of the function `fmincon` in MATLAB. This function finds a constrained minimum of a scalar function of several variables starting at initial estimates, in this case the minimum of $\sum_{t=1}^{T} \varepsilon_t^2 = \sum_{t=1}^{T} \left(\bar{y}_t - \widehat{\bar{y}}_t \right)^2$.

2.3. Dynamical BLUP prediction model

In order to handle cases where the additive genetic relationship matrix $\boldsymbol{A}_t$ departs from a unity matrix, we may use a linear mixed model and BLUP solutions. As a starting point we may note that the reaction norm model in Equations (2a,b), with use of $u'_{1,t} = u_{1,t} - u_{1,ref}$, etc., and $a_{i,t} - \bar{a}_t = a'_{i,t}$ etc., can be combined into

$$
\begin{aligned}
y_{i,t} = {} & \bar{y}_t + a_{i,t} - \bar{a}_t \\
& + \left(b_{1,i,t} - \bar{b}_{1,t} \right) u'_{1,t} + \cdots + \left(b_{q,i,t} - \bar{b}_{q,t} \right) u'_{q,t} \\
& + v_{i,t} + \eta_{1,i,t} u'_{1,t} + \cdots + \eta_{q,i,t} u'_{q,t} \\
= {} & \bar{y}_t + a'_{i,t} + b'_{1,i,t} u'_{1,t} + \cdots + b'_{q,i,t} u'_{q,t} \\
& + v_{i,t} + \eta_{1,i,t} u'_{1,t} + \cdots + \eta_{q,i,t} u'_{q,t}.
\end{aligned} \tag{6a}
$$

We thus use the individual variations $a'_{i,t} = a_{i,t} - \bar{a}_t$, $b'_{1,i,t} = b_{1,i,t} - \bar{b}_{1,t}$, $\ldots$, $b'_{q,i,t} = b_{q,i,t} - \bar{b}_{q,t}$ around the mean values $\bar{a}_t$, $\bar{b}_{1,t}$, $\ldots$, $\bar{b}_{q,t}$ as individual random effects. This leads to the linear mixed model for a population of N individuals,

$$
\boldsymbol{y}_t = \boldsymbol{X}\bar{y}_t + \widetilde{\boldsymbol{Z}}_t \begin{bmatrix} \boldsymbol{a}'_t \\ \boldsymbol{b}'_{1,t} \\ \vdots \\ \boldsymbol{b}'_{q,t} \end{bmatrix} + \boldsymbol{e}_t, \tag{6b}
$$

where $\boldsymbol{X} = \boldsymbol{1}_N$, $\widetilde{\boldsymbol{Z}}_t = \begin{bmatrix} 1 & u'_{1,t} & \cdots & u'_{q,t} \end{bmatrix} \bigotimes \boldsymbol{I}_N$, $\boldsymbol{a}'_t = \boldsymbol{a}_t - \boldsymbol{1}_N \bar{a}_t = \begin{bmatrix} a'_{1,t} & a'_{2,t} \cdots & a'_{N,t} \end{bmatrix}^T$, $\boldsymbol{b}'_{j,t} = \boldsymbol{b}_{j,t} - \boldsymbol{1}_N \bar{b}_{j,t} = \begin{bmatrix} b'_{j,1,t} & b'_{j,2,t} \cdots & b'_{j,N,t} \end{bmatrix}^T$, for $j = 1, 2, \cdots, q$, and $\boldsymbol{e}_t = \boldsymbol{v}_t + u'_{1,t} \boldsymbol{\eta}_{1,t} + \cdots + u'_{q,t} \boldsymbol{\eta}_{q,t}$. Here, $\boldsymbol{y}_t, \boldsymbol{a}'_t, \boldsymbol{b}'_{1,t}, \boldsymbol{b}'_{2,t}, \cdots, \boldsymbol{b}'_{q,t}, \boldsymbol{v}_t, \boldsymbol{\eta}_{1,t}, \boldsymbol{\eta}_{2,t}, \cdots, \boldsymbol{\eta}_{q,t}$ are $N \times 1$ vectors of individual values at time t, while $\bar{y}_t$ is the scalar valued fixed effect. The expected values $\mathrm{E}\left[\boldsymbol{a}'_t\right]$, $\mathrm{E}\left[\boldsymbol{b}'_{j,t}\right]$, $\mathrm{E}\left[\boldsymbol{v}_t\right]$ and $\mathrm{E}\left[\boldsymbol{\eta}_{j,t}\right]$ in this mixed model are zero by definition (Ch. 26, Lynch and Walsh, 1998; Robinson, 1991). From Equation (6b) follows the residual covariance matrix

$$
\boldsymbol{R}_t = E\left[\boldsymbol{e}_t \boldsymbol{e}_t^T\right] = \sigma_v^2 + u_1'^2 \sigma_{\eta_1}^2 + \cdots + u_q'^2 \sigma_{\eta_q}^2 = r_t. \tag{6c}
$$

In order to predict $\bar{y}_t$, $\boldsymbol{a}'_t$, and $\boldsymbol{b}'_{1,t}, \boldsymbol{b}'_{2,t}, \cdots \boldsymbol{b}'_{q,t}$ for any given parent generation, we may solve the BLUP equation (Ch. 26, Lynch and Walsh, 1998; Robinson, 1991; Henderson, 1950)

$$
\begin{bmatrix} \boldsymbol{X}^T \widetilde{\boldsymbol{R}}_t^{-1} \boldsymbol{X} & \boldsymbol{X}^T \widetilde{\boldsymbol{R}}_t^{-1} \widetilde{\boldsymbol{Z}}_t \\ \widetilde{\boldsymbol{Z}}_t^T \widetilde{\boldsymbol{R}}_t^{-1} \boldsymbol{X} & \widetilde{\boldsymbol{Z}}_t^T \widetilde{\boldsymbol{R}}_t^{-1} \widetilde{\boldsymbol{Z}}_t + \widetilde{\boldsymbol{G}}_t^{-1} \end{bmatrix} \begin{bmatrix} \widehat{\bar{y}}_t \\ \widehat{\boldsymbol{a}}'_t \\ \widehat{\boldsymbol{b}}'_{1,t} \\ \vdots \\ \widehat{\boldsymbol{b}}'_{q,t} \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}^T \widetilde{\boldsymbol{R}}_t^{-1} \boldsymbol{y}_t \\ \widetilde{\boldsymbol{Z}}_t^T \widetilde{\boldsymbol{R}}_t^{-1} \boldsymbol{y}_t \end{bmatrix},
$$

(6d)

where $\widetilde{\boldsymbol{G}}_t = \begin{bmatrix} G_{aa} & G_{ab} \\ \boldsymbol{G}_{ab}^T & \boldsymbol{G}_{bb} \end{bmatrix} \bigotimes \boldsymbol{A}_t = \boldsymbol{G} \bigotimes \boldsymbol{A}_t$, and $\widetilde{\boldsymbol{R}}_t = \boldsymbol{R}_t \bigotimes \boldsymbol{I}_N$. With $\boldsymbol{R}_t = r_t$ according to Equation (6c), this can be simplified into

$$
\begin{bmatrix} \boldsymbol{X}^T \boldsymbol{X} & \boldsymbol{X}^T \widetilde{\boldsymbol{Z}}_t \\ \widetilde{\boldsymbol{Z}}_t^T \boldsymbol{X} & \widetilde{\boldsymbol{Z}}_t^T \widetilde{\boldsymbol{Z}}_t + r_t \widetilde{\boldsymbol{G}}_t^{-1} \end{bmatrix} \begin{bmatrix} \widehat{\bar{y}}_t \\ \widehat{\boldsymbol{a}}'_t \\ \widehat{\boldsymbol{b}}'_{1,t} \\ \vdots \\ \widehat{\boldsymbol{b}}'_{q,t} \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}^T \boldsymbol{y}_t \\ \widetilde{\boldsymbol{Z}}_t^T \boldsymbol{y}_t \end{bmatrix}. \tag{6e}
$$

Note that this is a dynamical model in the sense that $\widetilde{\boldsymbol{Z}}_t$, $\widetilde{\boldsymbol{G}}_t$ and r_t vary with time.

With $\boldsymbol{u}' = \begin{bmatrix} u'_{1,t} & u'_{2,t} \cdots & u'_{q,t} \end{bmatrix}^T$, and since $\widetilde{\boldsymbol{G}}_t^{-1} = (\boldsymbol{G} \bigotimes \boldsymbol{A}_t)^{-1} = \boldsymbol{G}^{-1} \bigotimes \boldsymbol{A}_t^{-1}$, it follows from Equation (6e) that the BLUP equation for any given parent generation is

$$
[\bullet] \begin{bmatrix} \widehat{\bar{y}}_t \\ \widehat{\boldsymbol{a}}'_t \\ \widehat{\boldsymbol{b}}'_{1,t} \\ \vdots \\ \widehat{\boldsymbol{b}}'_{q,t} \end{bmatrix} = \begin{bmatrix} \boldsymbol{1}_N^T \boldsymbol{y}_t \\ \begin{bmatrix} 1 \\ \boldsymbol{u}' \end{bmatrix} \bigotimes \boldsymbol{I}_N \boldsymbol{y}_t \end{bmatrix} \tag{6f}
$$

where $[\bullet] =$

$$
\begin{bmatrix} N & \boldsymbol{1}_N^T \begin{bmatrix} 1 & \boldsymbol{u}'^T \end{bmatrix} \bigotimes \boldsymbol{I}_N \\ \begin{bmatrix} 1 \\ \boldsymbol{u}' \end{bmatrix} \bigotimes \boldsymbol{I}_N \boldsymbol{1}_N & \left(\begin{bmatrix} 1 \\ \boldsymbol{u}' \end{bmatrix} \begin{bmatrix} 1 & \boldsymbol{u}'^T \end{bmatrix} \right) \bigotimes \boldsymbol{I}_N + r_t \boldsymbol{G}^{-1} \bigotimes \boldsymbol{A}_t^{-1} \end{bmatrix}.
$$

Here, the predictions are unbiased in the sense that $\mathrm{E}\left[\widehat{\bar{y}}_t\right] = \mathrm{E}\left[\bar{y}_t\right]$ and $\mathrm{E}\left[\widehat{\boldsymbol{a}}'_t\right] = \boldsymbol{0}$, $\mathrm{E}\left[\widehat{\boldsymbol{b}}'_{1,t}\right] = \boldsymbol{0}$, etc. (Ch. 26, Lynch and Walsh, 1998; Robinson, 1991). From Equation (6f) follows the following theorem:

Theorem 1 *In Equation (6f), G_{aa} may be set to any fixed value, if at the same time the rest of the parameters in $\boldsymbol{G}$ as well as in r_t are given normalized values in relation to the value of G_{aa}.*

Proof : The term $r_t \boldsymbol{G}^{-1}$ in Equation (6f) remains constant if all elements in $\boldsymbol{G}$ and r_t are multiplied by the same factor.

Next, we must find how fitness affects the mean reaction norm trait predictions for the offspring of any given parent generation. As shown in Ergon (2023), and as verified by Theorem 2 below, the incremental changes in mean trait values can be found from the predicted random effects by use of Robertson's secondary theorem of natural selection (Ch. 6, Walsh and Lynch, 2018; Robertson, 1966), i.e.,

$$
\begin{bmatrix} \Delta \overline{a}_t \\ \Delta \overline{b}_{1,t} \\ \vdots \\ \Delta \overline{b}_{q,t} \end{bmatrix} = \begin{bmatrix} \mathrm{cov}\left(w_{i,t}, \widehat{a}'_{i,t}\right) \\ \mathrm{cov}\left(w_{i,t}, \widehat{b}'_{1,i,t}\right) \\ \vdots \\ \mathrm{cov}\left(w_{i,t}, \widehat{b}'_{q,i,t}\right) \end{bmatrix}, \tag{7}
$$

where $\widehat{a}'_{i,t}$, $\widehat{b}'_{1,i,t}$, ..., $\widehat{b}'_{q,i,t}$ are found from Equation (6f). A comparison with the multivariate breeder's Equation (3) indicates that Equation (7) assumes that $\boldsymbol{G} = \boldsymbol{P}$, which in general is not correct. However, as discussed in Ergon (2023), and verified by Theorem 2 below, this error is compensated by errors in the predicted random effects $\widehat{\boldsymbol{a}}'_t$, $\widehat{\boldsymbol{b}}'_{1,t}$, ..., $\widehat{\boldsymbol{b}}'_{q,t}$. The errors in the predicted random effects are thus just what is needed in order to find the correct predicted incremental changes according to Equation (7). With given initial mean trait values, Equation (7) gives $\overline{a}_t$, $\overline{b}_{1,t}$, ..., $\overline{b}_{q,t}$, and thus also $\overline{y}_t = \overline{a}_t + (\boldsymbol{u}_t - \boldsymbol{u}_{\mathrm{ref}})^T \overline{\boldsymbol{b}}_t$ according to Equation (2b), for use in the tuning model according to Figure 2. Initial values of the elements in $\overline{\boldsymbol{b}}_t$ for $t = 1$ must also here be found as part of the identification, while $\overline{a}_1$ follows from Equation (2b) by setting $\overline{y}_1 = 0$. See Section 3 for a simulation example, where BLUP estimates of $\boldsymbol{u}_{\mathrm{ref}}$, $\overline{\boldsymbol{b}}_1$, and the parameters in G and r_t are found by use of the function `fmincon` in MATLAB.

2.4. Differences regarding estimation of $\boldsymbol{u}_{\mathrm{ref}}$

Simulations show that estimation of $\boldsymbol{u}_{\mathrm{ref}}$ by means of the GRAD optimization method is difficult, while estimation by means of the BLUP optimization method is easier, at least with a single reference value. The fundamental difference is that the GRAD model makes use of $\boldsymbol{u}_{\mathrm{ref}}$ both in the computation of $\Delta \overline{a}_t$ and $\Delta \overline{\boldsymbol{b}}_t$ by use of Equation (5a), and in the computation of $\overline{y}_t = \overline{a}_t + \boldsymbol{u}'^T_t \overline{\boldsymbol{b}}_t$, while the BLUP model gives $\Delta \overline{a}_t$ and $\Delta \overline{\boldsymbol{b}}_t$ by use of Equation (7), without direct use of $\boldsymbol{u}_{\mathrm{ref}}$.

2.5. Comparison of prediction methods for $\boldsymbol{A}_t = \boldsymbol{I}_N$

Under the assumption that $\boldsymbol{A}_t = \boldsymbol{I}_N$, it follows from Theorem 2 in Ergon (2023) that the BLUP model described above leads to the mean trait prediction equation

$$
\begin{bmatrix} \Delta \overline{a}_t \\ \Delta \overline{b}_{1,t} \\ \vdots \\ \Delta \overline{b}_{q,t} \end{bmatrix} = [\bullet] \begin{bmatrix} 1 \\ \boldsymbol{u}' \end{bmatrix} \mathrm{cov}\left(w_{i,t}, y_{i,t}\right), \tag{8}
$$

where $[\bullet] = \boldsymbol{G} \left(\begin{bmatrix} 1 \\ \boldsymbol{u}' \end{bmatrix} \begin{bmatrix} 1 & \boldsymbol{u}'^T \end{bmatrix} \boldsymbol{G} + r_t \boldsymbol{I}_{q+1} \right)^{-1}$. A comparison of the GRAD and BLUP prediction models leads to the following theorem, which follows from Theorem 3 in Ergon (2023):

Theorem 2 *With $\boldsymbol{A}_t = \boldsymbol{I}_N$, the GRAD and BLUP models result in exactly the same mean reaction norm trait predictions, i.e., Equation (5a) and Equation (8) give identical results, independent of population size.*

2.6. Identifiability and prediction error issues

Ideally, optimization according to Figure 2 gives $\widehat{\overline{y}}_t \equiv \overline{y}_t$ for $t = 1$ to T, and estimated entities $\widehat{\boldsymbol{u}}_{\mathrm{ref}} \equiv \boldsymbol{u}_{\mathrm{ref}}$, $\widehat{\overline{a}}_1 \equiv \overline{a}_1$, $\widehat{\overline{\boldsymbol{b}}}_1 \equiv \overline{\boldsymbol{b}}_1$, $\widehat{\boldsymbol{G}} \equiv \boldsymbol{G}$, and $\widehat{\boldsymbol{P}} \equiv \boldsymbol{P}$. That would require an input $\boldsymbol{u}_t$ that is persistently exciting of sufficient order (Ch. 13, Ljung, 1999), a perfect tuning model with perfect measurements of the input and output signals, and that $T \to \infty$. In order for the system to be parameter identifiable (Ch. 6, Stoica and Söderström, 1989), it should in theory be possible to obtain $\widehat{\overline{y}}_t \equiv \overline{y}_t$ for $t = 1$ to T only with $\widehat{\boldsymbol{u}}_{\mathrm{ref}} \equiv \boldsymbol{u}_{\mathrm{ref}}$, $\widehat{\overline{a}}_1 \equiv \overline{a}_1$, $\widehat{\overline{\boldsymbol{b}}}_1 \equiv \overline{\boldsymbol{b}}_1$, $\widehat{\boldsymbol{G}} \equiv \boldsymbol{G}$, and $\widehat{\boldsymbol{P}} \equiv \boldsymbol{P}$, i.e., with no other parameter values. It should thus not be pe possible for an error in one parameter value to be compensated by errors in other parameter values.

It may be argued that $P_{yy,t}$ according to Equation (5b) includes the term $G_{aa} + \sigma_v^2$, which indicates a structural identifiability problem. This is, however, not the case when $\widehat{G}_{aa}$ is given a fixed value in accordance with Theorem 1.

The identifiability property of the GRAD system in Equations (2a) and (5a) is given by the following theorem:

Theorem 3 *The system given by Equation (2a) and the GRAD Equation (5a) is parameter identifiable, provided that $G_{aa} \neq 0$ exists, and that $\widehat{G}_{aa}$ is given a fixed value.*

See Appendix A for detailed proof.

It follows from Theorems 2 and 3 that also a BLUP system in the special case with a genetic relationship matrix $\boldsymbol{A}_t = \boldsymbol{I}_N$ is parameter identifiable. There seems to be no reason why the identifiability properties of a BLUP system should be affected when $\boldsymbol{A}_t \neq \boldsymbol{I}_N$, and this reasoning leads to the following conjecture, which is also supported by simulation results in Section 3:

Conjecture 1 *Theorem 3 is valid also for the system given in Equation* (2a) *and the BLUP Equation* (6f).

When the reference environment is fixed but incorrect, the situation is different. It follows from Equation (5a) that in order to minimize $\sum_{t=1}^{T} \left(\overline{y}_t - \widehat{\widehat{y}}_t \right)^2$, the effects of the error in $\widehat{\boldsymbol{u}}_{\mathrm{ref}}$ must somehow be compensated by errors in $\widehat{\boldsymbol{G}}$ or $\widehat{\boldsymbol{P}}$. The resulting deviations in parameter values may be large, and as shown by simulations in Section 3 this will result in errors in especially the predicted mean intercept value. As discussed in Ergon (2022a), the effects on the predicted mean slope values are smaller.

3. Simulation Results

I will here assume that true data is generated by a BLUP model, and test the GRAD and BLUP optimization methods in several simulation examples. In Example 1 we will see that the GRAD tuning model has problems regarding estimation of the environmental reference value, and that is the case also if true data is generated by a GRAD model. In Example 2 we will see that the BLUP tuning model gives good estimates of u_{ref}, and good predictions of the mean reaction norm traits, also with measurement errors. With perfect measurements, also the parameter estimates will be good. This example also shows that a fixed but incorrect value of u_{ref} may give large prediction errors, and it verifies that the BLUP and GRAD predictions with an additive genetic relationship matrix $\boldsymbol{A}_t = \boldsymbol{I}_N$ are identical (Theorem 2). Example 3 shows that BLUP optimization may give good mean trait predictions also with true data generated by use of time-varying parameters, but then with large errors in the parameter estimates. Finally, Example 4 verifies that the final value of the criterion function $\sum_{t=1}^{T} \left(\overline{y}_t - \widehat{\widehat{y}}_t \right)^2$ may be used for model selection (see discussion in Section 4).

3.1. Toy example, fitness function and input signals

For the purpose of testing the BLUP/PEM optimization method, I assume a true system with reaction norms according to Equations (2a,b), where $y_{i,t}$ is the individual clutch-initiation date for a certain bird

species. This is similar to the toy system in Ergon (2022a), except that a second input signal is added, and that I will assume a continuous clutch-initiation time (which may be more realistic). As in Ergon (2022a), the individual fitness is an integer number from 0 to 10 (the number of offspring).

The input signals are the present spring temperature $u_{1,\mathrm{t}} = u_{\mathrm{t}}$, and the spring temperature $u_{2,\mathrm{t}} = u_{\mathrm{t}-1}$ a generation back in time. The delayed effect could for example be food abundance in year t determined by the temperature in year $t-1$. This means that the two input signals have the same reference environment, and for simplicity I assume a temperature scale such that $u_{\mathrm{ref}} = 0$.

The individual fitness values are rounded values of

$$W_{i,t} = 10 \, \exp \left(-(y_{i,t} - \theta_t)^2 / 2\omega^2 \right), \qquad (9)$$

where θ_t is the phenotypic value that maximizes fitness, while $\omega^2 = 1000$ is the squared width of the Gaussian fitness function.

Assume a constant or slowly varying mean $\mu_{U,t}$ of a stochastic environment, with added white noise (iid zero mean normal random variations) $u_{n,t}$, with variance $\sigma_{U_n}^2 = 0.5$, i.e., $u_{\mathrm{t}} = \mu_{U,t} + u_{n,t}$. In a corresponding way assume that $\theta_t = \mu_{\Theta,t} + \theta_{n,t}$, where $\mu_{\Theta,t} = -20\mu_{U,t}$ and $\theta_{n,t}$ is white noise with variance $\sigma_{\Theta_n}^2 = 200$. Also assume that $u_{n,t}$ and $\theta_{n,t}$ are correlated with covariance $\mathrm{cov}\,(u_{1,t}, \theta_t) = -2.5$, as explained from fluctuations in the environment during the development of the individuals in the parent generation (Lande, 2009). Since u_{t} is white noise and $u_{2,\mathrm{t}} = u_{\mathrm{t}-1}$, we have $\mathrm{cov}\,(u_{2,t}, \theta_t) = 0$.

Further assume that u_t and θ_t are noisy ramp functions as shown in Figure 3, starting at $t = 10$. The choice of a negative trend in θ_t, and thus in $\overline{y}_t$, is inspired by common cases where a positive temperature trend leads to earlier breeding dates for various natural populations (e.g., Bowers et al., 2016). The noisy temperature trend in Figure 3 is similar to the registered yearly mean temperature in Oslo, Norway, from 1960 to 2020 (klimaservicesenter, 2023), using the temperature in 1960 as zero-point. In order to make the simulation examples more realistic, only data from $t = 31$ to 60 (1991 to 2020) are used for parameter estimation and mean reaction norm trait predictions. This will in most realizations mean that the reference environment is not within the range of input data used in the optimizations.

3.2. Population size, measurement errors, initial values, and result presentation

The GRAD and BLUP optimization algorithms are tested in four examples in Subsection 3.4 below, with
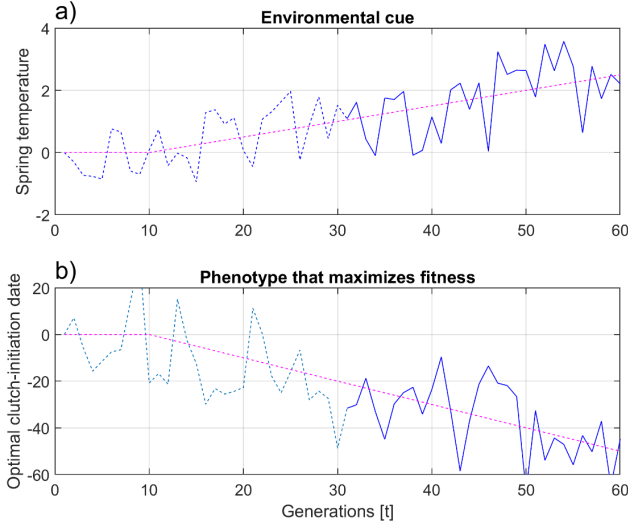
Figure 3: Noisy ramp functions u_t (spring temperature, panel a) and θ_t (optimal clutch initiation date, panel b), with $u_{\text{ref}} = 0$, $\mu_{\Theta,t} = -20\mu_{U,t}$, $\sigma^2_{U_n} = 0.5$, $\sigma^2_{\Theta_n} = 200$ and $\text{cov}(u_{1,t}, \theta_t) = -2.5$. The solid parts of the curves indicate data used for parameter estimation and mean reaction norm trait predictions.

in all 10 special cases. The population size is in general $N = 100$, although for comparison a test with $N = 10,000$ is included in Appendix B. The algorithms are tested with perfect input and output information, as well as with measurement errors in $u_{1,t}$, $u_{2,t}$ and $y_{i,t}$. The errors in temperature measurements are implemented by adding normally distributed white noise with standard deviations 0.1 to $u_{1,t}$ and $u_{2,t}$, which is 14% of the standard deviation in the yearly variations. The errors in individual phenotypes are implemented as uniformly distributed random errors in the interval -0.5 to 0.5, i.e., errors up to half a day in clutch-initiation time. The relative individual fitness $\boldsymbol{w}_t$, i.e., the relative number of offspring, will be assumed to be correctly counted.

All results are found by minimization of $\sum_{t=31}^{60} \varepsilon_t^2 = \sum_{t=31}^{60} \left(\overline{y}_t - \widehat{\overline{y}}_t \right)^2$ by use of the function `fmincon` in MATLAB. In all examples the true value $G_{aa} = 5$ was used, such that estimates of the other $\boldsymbol{G}$ and $\boldsymbol{P}$ parameters are found relative to G_{aa} (Theorem 1). The initial values for these parameters were sat to 0. The initial mean trait values were sat to $\widehat{\overline{b}}_{1,init} = 0$ and $\widehat{\overline{b}}_{2,init} = 1$ (because the true value is 0), while $\widehat{\overline{a}}_{init}$ follows from Equation (2b) with $\widehat{\overline{y}}_{init}$ set to zero. In cases where $\widehat{u}_{\text{ref}}$ is a free variable, the initial value was sat to 1 (because the true value is 0).

The GRAD optimization with population size $N = 100$ takes around 1 second on an HP EliteBook $\times 360$ 1030 $G3$ laptop. With $N = 100$, the optimization time for the BLUP optimization increased to around 400 seconds, owing to the repeated inversions of the $(3N + 1) \times (3N + 1)$ matrix in Equation (6f). No attempts were made to speed up this optimization.

Simulation results are presented as plots over predicted mean values $\widehat{\overline{y}}_t$, $\widehat{\overline{a}}_t$, $\widehat{\overline{b}}_{1,t}$ and $\widehat{\overline{b}}_{2,t}$, as compared to true mean values $\overline{y}_t$, $\overline{a}_t$, $\overline{b}_{1,t}$ and $\overline{b}_{2,t}$. Estimated parameter values are given in tables as mean values and standard errors, $Mean \pm SE$, based on 10 repeated simulations with different realizations of random inputs (in Case 6 increased to 100). The relative errors in total change of predictions over 30 generations are included in the tables, computed as, for example, $\Delta_{30}^{\text{error}} \widehat{\overline{a}}_t\% = 100 \left(\Delta_{30} \widehat{\overline{a}}_t - \Delta_{30} \overline{a}_t \right) / \Delta_{30} \overline{a}_t$ etc., where $\Delta_{30} \widehat{\overline{a}}_t = \widehat{\overline{a}}_{60} - \widehat{\overline{a}}_{31}$ and $\Delta_{30} \overline{a}_t = \overline{a}_{60} - \overline{a}_{31}$. The final values $\sum \varepsilon_{t,final}^2$ of $\sum_{t=31}^{60} \varepsilon_t^2$ are also included, as they may possibly be used for model selection.

3.3. Generation of true BLUP data

True mean responses $\overline{y}_t$, $\overline{a}_t$, $\overline{b}_{1,t}$ and $\overline{b}_{2,t}$ are generated by means of the dynamical BLUP model (6f) and the prediction Equation (7), the fitness function in Equation (9), and known parameter values in the covariance matrices

$$\boldsymbol{G} = \begin{bmatrix} G_{aa} & G_{ab_1} & 0 \\ G_{ab_1} & G_{b_1b_1} & 0 \\ 0 & 0 & G_{b_2b_2} \end{bmatrix} = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}$$

and

$$\boldsymbol{P} = \begin{bmatrix} G_{aa} + \sigma_v^2 & G_{ab_1} & 0 \\ G_{ab_1} & G_{b_1b_1} + \sigma_{\eta_1}^2 & 0 \\ 0 & 0 & G_{b_2b_2} + \sigma_{\eta_2}^2 \end{bmatrix}$$
$$= \begin{bmatrix} 10 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Here, $G_{ab_1} = 0$ in the true system, but left as a free parameter in the optimizations. It is assumed that the population is fully adapted to the stationary stochastic environment up to $t = 10$ (1970), i.e., to the expected spring temperature $\mathrm{E}[u_t] = 0$ (Ergon, 2022a). From this follows that $G_{ab_1} = G_{ab_2} = 0$ (Lande, 2009)), and that the optimal mean plasticity slope values in a stationary stochastic environment are $\overline{b}_{1,init,true} = \text{cov}(u_{1,t}, \theta_t)/\sigma_{U_n}^2 = -5$, and $\overline{b}_{2,init,true} = \text{cov}(u_{2,t}, \theta_t)/\sigma_{U_n}^2 = 0$ (Lande, 2009; Ergon and Ergon, 2017). This will thus be the initial mean plasticity slope values at generation 1.

For simplicity, the additive genetic relationship matrix is assumed to be step-wise time varying, with a structure as shown in Equation (10) (for population size $N = 6$),

$$\boldsymbol{A}_t = \boldsymbol{I}_N + c \begin{bmatrix} 0 & 1/2 & 1/4 & 1/4 & 1/4 & 1/4 \\ 1/2 & 0 & 1/2 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/2 & 0 & 1/2 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/2 & 0 & 1/2 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/2 & 0 & 1/2 \\ 1/4 & 1/4 & 1/4 & 1/4 & 1/2 & 0 \end{bmatrix} \quad (10)$$

where $c = 0$ for $0 < t \leq 30$, $c = 0.25$ for $30 < t \leq 40$, $c = 0.5$ for $40 < t \leq 50$, and $c = 1$ for $50 < t \leq 60$. All individuals are thus unrelated in the first 30 generations, while in the last 10 generations every individual in the population has one sibling, while the rest are cousins. Note that the 30 generations that are used in the model tuning according to Figure 2, have ten generations with $c = 0.25$, ten generations with $c = 0.5$, and ten generations with $c = 1$. This is an unrealistic example of a dynamical relationship matrix, but it serves the purpose of a test matrix for the BLUP/PEM method.

New random effects were at each generation found as $\boldsymbol{z}_t = \sqrt[M]{\boldsymbol{A}_t} \boldsymbol{z}_{0,t}$, where $\boldsymbol{z}_t$ stands for $\boldsymbol{a}'_t$, $\boldsymbol{b}'_{1,t}$, or $\boldsymbol{b}'_{2,t}$, and where $\sqrt[M]{\boldsymbol{A}}$ is the matrix square root, i.e., $\sqrt[M]{\boldsymbol{A}_t} \sqrt[M]{\boldsymbol{A}_t} = \boldsymbol{A}_t$. The different data vectors $\boldsymbol{z}_{0,t}$ were drawn from normal distributions in accordance with the given $\boldsymbol{G}$ matrix. For the three values of the factor c in Equation (10) used in generations 31 to 60, this reduced the variances of the random effects to approximately 95%, 90% and 75% of the nominal values, respectively. The random effects were also mean centred.

3.4. Simulation examples

Example 1: BLUP true data with GRAD prediction model

Results for different cases with a BLUP true data model and a GRAD tuning model, are given in Table 1. Note that also with perfect measurements (Case 1), the value of u_{ref} is estimated with a large standard error, resulting in a large SE value for $\Delta_{30}^{\text{error}} \widehat{a}_t$. This is to some extent understandable, since the true data is generated by a BLUP model, but also with true data generated by a GRAD model we find the poor results $\widehat{u}_{\text{ref}} = -0.07 \pm 0.29$. With use of the known value of u_{ref} (Case 2), the parameters $G_{b_1 b_1}$ and $G_{b_2 b_2}$ are estimated with small errors, while the estimated white noise variances, $\widehat{\sigma}_v^2$, $\widehat{\sigma}_{\eta_1}^2$ and $\widehat{\sigma}_{\eta_2}^2$ have large errors, owing to the modeling error. The mean reaction norm traits are in this case predicted with SE values up to

5%., i.e., GRAD/PEM finds good predictions at the price of large errors in the estimated noise variances. The known value of u_{ref} combined with measurement errors (Case 3), gives SE values up to 11%. With a factor five increase in the measurement errors, the mean trait SE values were up to 40%, without any indication of bias.

Example 2: BLUP true data with BLUP prediction model

Simulation results with parameter estimates and mean trait predictions found by use of the dynamical BLUP algorithm, are shown in Table 2 and Figure 4. It is interesting to note that the environmental reference value is estimated far better than with use of the GRAD algorithm, for reasons as explained in Subsection 2.4. With perfect measurements (Case 4), the result is improved to $\widehat{u}_{\text{ref}} = 0.001 \pm 0.002$, and as a result the parameter estimates without measurement errors are close to perfect. With measurement errors (Case 5), the BLUP results for $\widehat{u}_{\text{ref}}$ with $N = 100$ are in fact just as good as for $N = 10,000$ with use of true GRAD data and GRAD optimization (Appendix B). Measurement errors result in large errors in some of the parameters, with consequences as discussed in Section 4, but the mean trait SE values are still limited to 8%. With a factor five increase in the measurement errors, the mean trait SE values increased to around 20%, still without any indication of bias. When in addition to measurement errors the incorrect relationship matrix $\boldsymbol{A}_t = \boldsymbol{I}_N$ is used in the BLUP optimization algorithm (Case 6), the mean trait predictions are still quite good, but the price to pay is large errors in the parameter estimates.

Table 2 also shows mean trait results for GRAD predictions based on the BLUP parameter results, i.e., when the genetic relationships are ignored. For Case 4 and 5, this leads to around 10% overestimation of the mean trait changes. For Case 6, i.e., with BLUP parameter estimates based on $\boldsymbol{A}_t = \boldsymbol{I}_N$, the GRAD and BLUP predictions are identical, as follows from Theorem 2, and this is the case for population sizes down to $N = 2$.

The results in Table 2 assumes estimated values of the reference environment. If we instead use fixed values of $\widehat{u}_{\text{ref}}$, we obtain results as in Table 3. Note the large values of $\Delta_{30}^{\text{error}} \widehat{a}_t \%$ for $\widehat{u}_{\text{ref}} \neq u_{\text{ref}}$, while as discussed in Ergon (2022a) the effects on the predicted mean slope values are smaller. Also note the large prediction errors for $\widehat{u}_{\text{ref}} = 1$, i.e., for approximately the initial value of the available temperature data, which it may be tempting to use as a reference.

Since $\boldsymbol{A}_t \neq \boldsymbol{I}_N$ according to Equation (10) reduces the variances of the additive genetic effects with 5 to 25%, for the different values of the factor c, it must

Table 1: Estimation and prediction results with BLUP true data model and GRAD tuning model. Results are based on 10 simulations with different realizations of all random input variables. The population size was $N = 100$.
Case 1: Perfect values of $u_{1,t}$, $u_{2,t}$ and $y_{i,t}$, and $\widehat{u}_{\mathrm{ref}}$ as free variable.
Case 2: Perfect values of $u_{1,t}$, $u_{2,t}$ and $y_{i,t}$, and $\widehat{u}_{\mathrm{ref}} = u_{\mathrm{ref}} = 0$.
Case 3: Measurement errors in $u_{1,t}$, $u_{2,t}$ and $y_{i,t}$, and $\widehat{u}_{\mathrm{ref}} = u_{\mathrm{ref}} = 0$.

Parameter	True value	Optimization results Case 1	Optimization results Case 2	Optimization results Case 3
$\widehat{G}_{b_1 b_1}$	1	1.01 ± 0.30	0.99 ± 0.07	1.00 ± 0.23
$\widehat{G}_{b_2 b_2}$	0.5	0.55 ± 0.07	0.45 ± 0.08	0.51 ± 0.19
$\widehat{G}_{ab_1}$	0	0.12 ± 0.46	-0.22 ± 0.36	-0.03 ± 0.85
$\widehat{\sigma}_v^2$	5	3.67 ± 2.21	3.52 ± 1.14	4.05 ± 2.26
$\widehat{\sigma}_{\eta_1}^2$	1	1.50 ± 0.66	0.86 ± 0.41	1.48 ± 0.39
$\widehat{\sigma}_{\eta_2}^2$	0.5	1.11 ± 0.84	0.57 ± 0.48	0.82 ± 0.63
$\widehat{b}_{1,init}$	–	-5.10 ± 0.07	-5.12 ± 0.04	-5.08 ± 0.06
$\widehat{b}_{2,init}$	–	-0.05 ± 0.03	-0.08 ± 0.05	-0.07 ± 0.06
$\widehat{u}_{\mathrm{ref}}$	0	-0.19 ± 0.67	0	0
$\sum \varepsilon_{t,final}^2$	–	$(100 \pm 50)10^{-4}$	$(176 \pm 160)10^{-4}$	$(236 \pm 77)10^{-4}$
$\Delta_{30}^{\mathrm{error}} \widehat{a}_t \%$	–	-8 ± 34	3 ± 3	0 ± 7
$\Delta_{30}^{\mathrm{error}} \widehat{b}_{1,t} \%$	–	-2 ± 3	-2 ± 2	-2 ± 4
$\Delta_{30}^{\mathrm{error}} \widehat{b}_{2,t} \%$	–	-4 ± 14	-4 ± 5	0 ± 11

Table 2: Estimation and prediction results by use of the BLUP optimization algorithm. Results are based on 10 simulations with different realizations of all random input variables. The population size was $N = 100$, and $\widehat{u}_{\mathrm{ref}}$ is in all cases a free variable.
Case 4: Perfect values of $u_{1,t}$, $u_{2,t}$ and $y_{i,t}$, with $\boldsymbol{A} \neq \boldsymbol{I}_N$ used in the tuning model.
Case 5: As Case 4, but with measurement errors in $u_{1,t}$, $u_{2,t}$ and $y_{i,t}$.
Case 6: As Case 5, but with $\boldsymbol{A} = \boldsymbol{I}_N$ used in the BLUP tuning model, and number of realizations increased to 100.

| Parameter	True value	Case 4	Case 5	Case 6
$\widehat{G}_{b_1 b_1}$	1	1.0000 ± 0.0000	1.04 ± 0.10	0.85 ± 0.29
$\widehat{G}_{b_2 b_2}$	0.5	0.5000 ± 0.0000	0.47 ± 0.07	0.56 ± 0.23
$\widehat{G}_{ab_1}$	0	0.0001 ± 0.0002	-0.08 ± 0.40	0.48 ± 1.27
$\widehat{\sigma}_v^2$	5	5.0002 ± 0.0004	4.93 ± 1.30	4.87 ± 2.31
$\widehat{\sigma}_{\eta_1}^2$	1	1.0000 ± 0.0000	1.06 ± 0.25	1.71 ± 1.00
$\widehat{\sigma}_{\eta_2}^2$	0.5	0.5000 ± 0.0001	0.43 ± 0.21	0.97 ± 0.79
$\widehat{b}_{1,init}$	–	-5.10 ± 0.07	-5.09 ± 0.09	-5.10 ± 0.08
$\widehat{b}_{2,init}$	–	-0.05 ± 0.04	-0.05 ± 0.07	-0.06 ± 0.04
$\widehat{u}_{\mathrm{ref}}$	0	-0.001 ± 0.002	-0.003 ± 0.004	0.002 ± 0.086
$\sum \varepsilon_{t,final}^2$	–	$(7 \pm 19)10^{-11}$	$(17 \pm 6)10^{-3}$	$(279 \pm 103)10^{-4}$
$\Delta_{30}^{\mathrm{error}} \widehat{a}_t \%$	–	0 ± 0	0 ± 4	3 ± 7
$\Delta_{30}^{\mathrm{error}} \widehat{b}_{1,t} \%$	–	0 ± 0	0 ± 3	-2 ± 7
$\Delta_{30}^{\mathrm{error}} \widehat{b}_{2,t} \%$	–	0 ± 0	0 ± 8	-3 ± 9
$\Delta_{30}^{GRAD} \widehat{a}_t \%$	–	9 ± 1	9 ± 5	3 ± 7
$\Delta_{30}^{GRAD} \widehat{b}_{1,t} \%$	–	11 ± 1	10 ± 4	-2 ± 7
$\Delta_{30}^{GRAD} \widehat{b}_{2,t} \%$	–	11 ± 1	10 ± 9	-3 ± 9
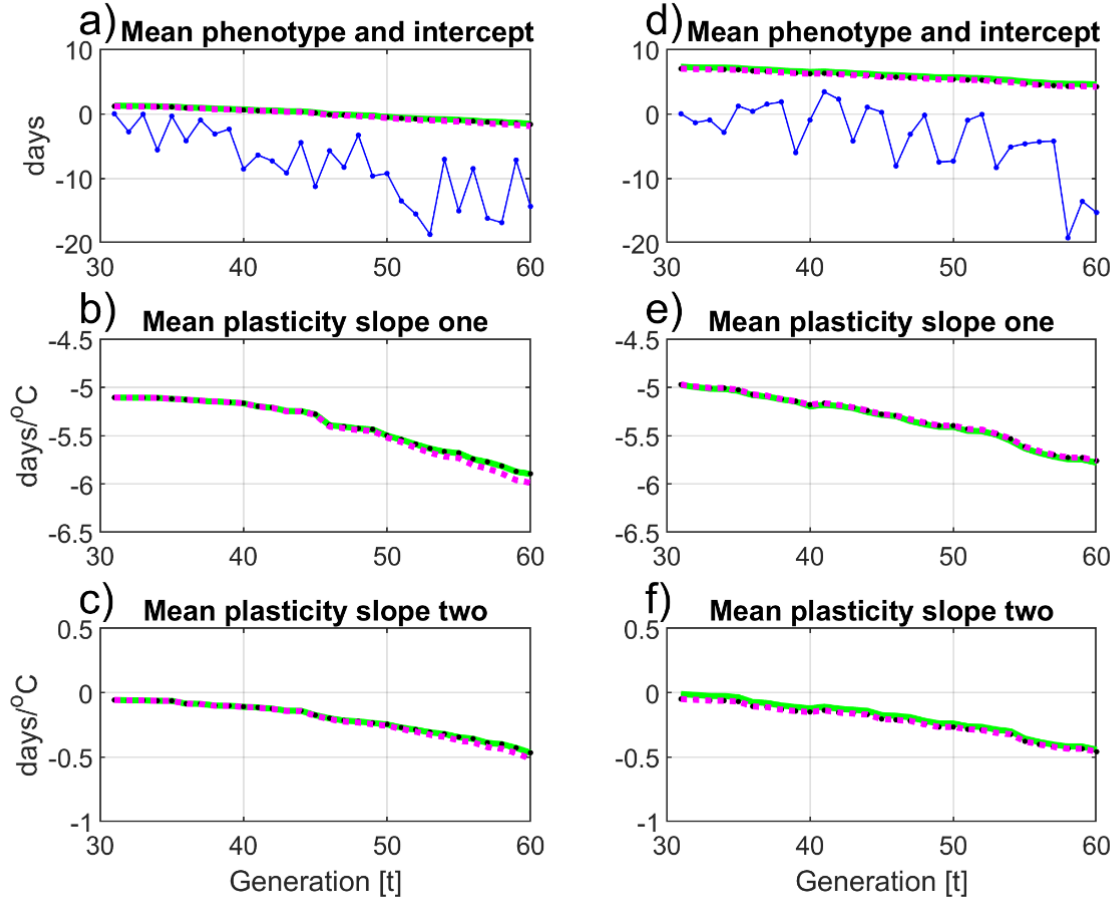
Figure 4: Simulation results for BLUP system. Panels a), b) and c) show typical results for Case 4 in Table 2, with $\boldsymbol{A}_t$ as given in Equation (10) (although for population size $N = 100$). Panels d), e) and f) show typical results for Case 6, with $\boldsymbol{A} = \boldsymbol{I}_N$ used in the BLUP optimization procedure. True $\overline{y}_t$ values are shown by solid blue lines, while final predictions $\widehat{\overline{y}}_t$ are shown by blue dots. True $\overline{a}_t$, $\overline{b}_{1,t}$ and $\overline{b}_{2,t}$ responses are shown by green lines, while BLUP predictions $\widehat{\overline{a}}_t$, $\widehat{\overline{b}}_{1,t}$ and $\widehat{\overline{b}}_{2,t}$ are shown by black dots. The corresponding GRAD predictions using BLUP parameters are shown by dotted magenta lines.

Table 3: Errors in predicted total relative change of $\overline{a}_t$, $\overline{b}_{1,t}$ and $\overline{b}_{2,t}$ over 30 generations as for Case 4 in Table 2, except for fixed reference environments used in the BLUP optimization procedure.

$\widehat{u}_{\mathrm{ref}}$	$\Delta_{30}^{\mathrm{error}}\widehat{\overline{a}}_t\%$	$\Delta_{30}^{\mathrm{error}}\widehat{\overline{b}}_{1,t}\%$	$\Delta_{30}^{\mathrm{error}}\widehat{\overline{b}}_{2,t}\%$
-0.5	-23 ± 1	0 ± 2	1 ± 3
-0.2	-9 ± 3	1 ± 1	1 ± 1
0	0 ± 0	0 ± 0	0 ± 0
0.2	10 ± 0	-1 ± 2	-2 ± 2
0.5	25 ± 3	-9 ± 5	-9 ± 4
1	52 ± 3	-29 ± 6	-31 ± 11

Table 4: Estimation and prediction results as in Table 2, but with G_{aa} increasing linearly from 5 to 10 over 60 generations.
Case 7: Results found by use of $\widehat{G}_{aa} = 5$ and otherwise true parameter values.
Case 8: Results found by minimization of $\sum_{t=31}^{60} \varepsilon_t^2$, without measurement errors.
Case 9: Results found by minimization of $\sum_{t=31}^{60} \varepsilon_t^2$, with measurement errors

Parameter	True values	Case 7	Case 8	Case 9
$\widehat{G}_{b_1 b_1}$	1	1	0.49 ± 0.04	0.43 ± 0.26
$\widehat{G}_{b_2 b_2}$	0.5	0.5	0.32 ± 0.02	0.33 ± 0.07
$\widehat{G}_{ab_1}$	0	0	0.28 ± 0.15	0.48 ± 0.61
$\widehat{\sigma}_v^2$	5	5	3.84 ± 0.27	4.24 ± 1.02
$\widehat{\sigma}_{\eta_1}^2$	1	1	0.54 ± 0.11	0.62 ± 0.34
$\widehat{\sigma}_{\eta_2}^2$	0.5	0.5	0.25 ± 0.11	0.22 ± 0.21
$\widehat{\overline{b}}_{1,init}$	–	–	-5.11 ± 0.06	-5.07 ± 0.08
$\widehat{\overline{b}}_{2,init}$	–	–	-0.05 ± 0.04	-0.07 ± 0.04
$\widehat{u}_{\text{ref}}$	0	0	0.002 ± 0.015	-0.003 ± 0.024
$\sum \varepsilon_{t,final}^2$	–	–	$(18 \pm 18)10^{-4}$	$(218 \pm 68)10^{-4}$
$\Delta_{30}^{\text{error}}\widehat{\overline{a}}_t \%$	–	-33 ± 1	-1 ± 1	-2 ± 5
$\Delta_{30}^{\text{error}}\widehat{\overline{b}}_{1,t} \%$	–	16 ± 1	2 ± 2	4 ± 7
$\Delta_{30}^{\text{error}}\widehat{\overline{b}}_{2,t} \%$	–	17 ± 1	2 ± 2	-1 ± 9

be expected that the responses on the gradual change in environment are slowed down. As a test, this effect on the BLUP results were therefore investigated by recording the true change in intercept value from generation 31 to generation 60, with use of the same set of typical u_t and θ_t input data in 100 repeated simulations. For $\boldsymbol{A}_t = \boldsymbol{I}_N$ in the true model, the results was $\Delta_{30}\overline{a} = -3.4 \pm 0.1$ days, while $\boldsymbol{A}_t \neq \boldsymbol{I}_N$ gave $\Delta_{30}\overline{a} = -2.9 \pm 0.1$ days, which shows that the responses are indeed slowed down.

Example 3: BLUP true data with time-varying $\boldsymbol{G}$ matrix

Results with a true system as in Example 2, but with a parameter value G_{aa} that increases linearly from $G_{aa,1} = 5$ to $G_{aa,60} = 10$, are shown in Table 4. Mean trait predictions based on $\widehat{G}_{aa} = 5$, and otherwise true parameter values, will now underestimate the total change $\Delta_{30}\overline{a}_t$ (Case 7). However, predictions based on the constant value $\widehat{G}_{aa} = 5$ and otherwise optimized parameter values give good mean trait predictions (Case 8), because parameter values are found such that the variation in G_{aa} is very much compensated for. When Case 8 is repeated with measurement errors (Case 9), the mean trait predictions are still fairly good, although the SE values increase markedly. In this case also the SE values for the parameter estimates increase markedly. In Cases 8 and 9 some of the estimated parameter values are far from the true values.

Example 4: BLUP true data with reduced BLUP prediction model

In an additional case, Case 10, the BLUP tuning model was reduced by setting $\widehat{G}_{b_2 b_2} = \widehat{\sigma}_{\eta_2}^2 = 0$, while it at the same time it had to handle measurement noise as in Case 5. This resulted in $\widehat{u}_{\text{ref}} = 0.03 \pm 0.1$, $\sum \varepsilon_{t,final}^2 = 0.32 \pm 0.21$, $\Delta_{30}^{\text{error}}\widehat{\overline{a}}_t \% = 35 \pm 17$ and $\Delta_{30}^{\text{error}}\widehat{\overline{b}}_{1,t} \% = 3 \pm 19$, i.e., markedly increased errors in the predicted changes of mean intercept. Note the large increase in $\sum \varepsilon_{t,final}^2$, as compared to Case 5, and this was the case for all realizations, although to various degree. This indicates that the value of $\sum \varepsilon_{t,final}^2$ may possibly be used as a tool for model selection (see discussion in Section 4).

4. Summary and discussion

In summary, there are two main purposes with this article. One is to show that system identification methods borrowed from the field of engineering control may play an important role in the study of evolutionary biological systems. The second is to show that the prediction error method (PEM) can be combined with the best linear unbiased predictions (BLUP) method, which is extensively used in domestic breeding programs. The theoretical foundation is the fact that a

reaction norm model can be formulated as a linear mixed model, where the random effects are the individual variations in intercept and plasticity slope values, while the environmental cues are included in a dynamical incidence matrix $\widetilde{\boldsymbol{Z}}_t$. From this follows the additional observation that restricted maximum likelihood (REML) based on data from a single generation cannot be used for parameter estimation, for the simple reason that each scalar element in the residual covariance matrix in the BLUP model includes several residual variances. In the formulation of the linear mixed model, we assumed a MISO system, with linear reaction norms as functions of environmental cues $u_{1,t}$, $u_{2,t}$, etc., but there is nothing in the theory that prevents us from use of non-linear reaction norms that are also functions of $u_{1,t}u_{2,t}$, $u_{1,t}^2$, $u_{2,t}^2$, etc., as discussed in Ergon (2018). A more general model for a MIMO system is developed in Ergon (2023).

Two evolutionary models for use in PEM as illustrated in Figure 2 was presented and developed in Section 2. The selection gradient (GRAD) model assumes that the additive genetic relationship matrix is a unity matrix, $\boldsymbol{A}_t = \boldsymbol{I}_N$, which implies that all individuals in the population are genetically unrelated. The BLUP model, on the other hand, can handle cases with inbreeding in the populations, i.e., with $\boldsymbol{A}_t \neq \boldsymbol{I}_N$. In both the GRAD/PEM and BLUP/PEM methods, the additive genetic variance G_{aa} can be set to any value, and the rest of the elements in the $\boldsymbol{G}$ and $\boldsymbol{P}$ matrices estimated in relation to that value (Theorem 1). When $\boldsymbol{A}_t = \boldsymbol{I}_N$, the GRAD and BLUP optimization algorithms give identical results (Theorem 2). It is shown that the GRAD prediction model is identifiable (Theorem 3), and from Theorem 2 thus follows that the BLUP prediction model is identifiable when $\boldsymbol{A}_t = \boldsymbol{I}_N$. It is conjectured that the BLUP model is identifiable also when $\boldsymbol{A}_t \neq \boldsymbol{I}_N$, and this conjecture is supported by simulation results.

Although the theory is limited to MISO systems, an extension of the GRAD algorithm to multiple-input multiple-output (MIMO) systems is demonstrated in Ergon (2022b). Such extensions are possible also for the BLUP algorithm (Ergon, 2023). The applicability of PEM in an evolutionary context is, however, not limited to the algorithms discussed, it should in principle work for any type of parametrized prediction model that includes mean reaction norm traits, or other latent variables.

A main theoretical result is that the environmental reference must be known or estimated, as also pointed out in Ergon (2022a). This was recognized by Scheiner et al. (2020), who in their simulations used "a 2000 generation equilibration period, with the mean phenotypic optimum held constant at 0, to allow the population

to reach mutation–selection–drift equilibrium". An important practical result is that the BLUP/PEM optimization method is far better than the GRAD/PEM method when it comes to estimation of the reference environment, at least with a single reference environment.

Independent of optimization method, there is an important principal difference between the traditional BLUP/REML approach, and the proposed GRAD/PEM and BLUP/PEM approaches. In the BLUP/REML method there is no feedback loop via a minimization algorithm as shown in Figure 2, where data from all observed generations are used. Instead, parameters in the $\boldsymbol{G}$ and $\boldsymbol{P}$ matrices are estimated by use of REML for a given base population with an assumed genetic relationship matrix $\boldsymbol{A}_t = \boldsymbol{I}_N$ (Ch. 19, Walsh and Lynch, 2018), or alternatively a relationship matrix computed from genomic information, and predictions of mean trait responses are then found forward in time in an open-loop fashion. PEM is a closed-loop method, where parameters are estimated, and mean trait responses predicted by use of information at all generations. It is, however, necessary to know the additive genetic relationship matrix $\boldsymbol{A}_t$ for all generations, which requires some assumptions regarding the first generation in a study. The assumption of a perfectly known pedigree may also then be unrealistic, and kinship estimation based on molecular markers might be a better alternative (Goudet et al., 2018). It is beyond the scope of this article to discuss parameter estimation by application of REML on Equation (6e). It is however obvious that the various components of r_t in Equation (6c) cannot be found by REML at a single generation, for the simple reason that only r_t appears in Equation (6e).

It should be noted that the primary goal of BLUP/PEM is not to find correct $\boldsymbol{G}$ and $\boldsymbol{P}$ parameter values as such, but to minimize errors in predictions of the mean phenotypic values $\widehat{\overline{y}}_t = \widehat{\overline{a}}_t + (\boldsymbol{u}_t - \widehat{\boldsymbol{u}}_{\mathrm{ref}})^T \widehat{\overline{\boldsymbol{b}}}_t$, which requires that the errors in the predicted mean reaction norm trait values $\widehat{\overline{a}}_t$ and $\widehat{\overline{\boldsymbol{b}}}_t$ are minimized. As further discussed below, this raises questions regarding model validation. With a correct model, perfect measurements, and long enough data series, however, PEM also leads to near perfect parameter estimates (Table 2, Case 4).

Hadfield et al. (2010) and other authors are critical to the use of BLUP for natural populations, for the main reason that the estimated variances of the individual random variables have large errors. As shown in Ergon (2022c, 2023), and as utilized in Equation (7) and verified by Theorem 2, these errors are just what is needed in order to find the correct mean trait predictions by means of Robertson's secondary theorem

of natural selection. Hadfield et al. (2010) also suggest that a good test of evolutionary change would be to simply compare the mean trait values of the first and final generations, as done in this paper.

The GRAD/PEM and BLUP/PEM optimization algorithms have been tested by simulations of an intercept-slope reaction norm toy system in Section 3, and these tests illustrate the feasibility of PEM in the microevolutionary context. The model includes two environmental inputs, first a noisy ramp function in the present year's temperature over 60 years (1961 - 2020), similar to the recorded temperature trend in Oslo, Norway, and second the last year's temperature. Corresponding correlated changes in the individual phenotypic value that maximizes fitness are used in the individual fitness function (Figure 3). Only data from the last 30 generations (1991 - 2020) are used in the optimizations, i.e., the reference temperature before 1960 is not included in the data. Although the optimal clutch-initiation time is forwarded by 30 days over 30 years (Figure 3), the true phenotypic changes is only 10 to 15 days (Figure 4). The phenotypic response is thus lagging behind the optimal response, which is typical for ramp responses of dynamical systems with time constants. The change in mean intercept over 30 generations is typically -3 days.

The BLUP results in Figure 4, panels a), b), and c), are based on random effects that are drawn from distributions that are affected by the matrix square root $\sqrt[M]{\boldsymbol{A}_t}$, with the structure of $\boldsymbol{A}_t$ given in Equation (10), which reduces the variances of the random effects up to 75% of the nominal values. As a result of this, the phenotypic response has a tendency of lagging further behind the optimal response.

It should be noted that the simulated MISO system with two inputs has a single and common reference environment u_{ref}, and that there is a major difference between the two algorithms when it comes to estimation of this reference environment. The GRAD optimization algorithm gives large errors in $\widehat{u}_{\text{ref}}$, and correspondingly large errors in the mean trait predictions (Table 1, Case 1), while the BLUP algorithm gives good estimates of u_{ref} (Subsection 2.4 and Table 2). When the GRAD algorithm is applied on data from small populations it may therefore be best to set $\widehat{u}_{\text{ref}}$ to a fixed value, based on a judgement on which environment the population is adapted to in the time period before environmental changes. This may also be the case for MISO and MIMO systems with a multivariate reference environment, although such cases need further invetigation. As shown in Table 3, large errors in $\widehat{u}_{\text{ref}}$ give large errors also in BLUP predictions, and in more realistic cases the estimated reference environment values must therefore be used with care. An alternative choice would be to use the initial environmental value in a recorded time series, but as shown in Table 3 that may be a bad idea.

Table 1, Case 2, and Table 2, Case 5, show results with measurement errors in the environmental and individual phenotypic data, and in these cases some of the parameters are estimated with large errors. The changes in mean trait values over 30 generations are still predicted without bias, and with standard errors less than 8%.

Table 2, Case 6, shows results with measurement errors and $\boldsymbol{A}_t = \boldsymbol{I}_N$, i.e., with a large error in the additive genetic relationship matrix, and the changes in mean trait values over 30 generations were still predicted with mean errors less than 3%, and standard errors less than 10%. The reason for these good predictions, in spite of the error in $\boldsymbol{A}_t$, is that minimization of $\sum_{t=1}^{30} \left(\overline{y}_t - \widehat{\overline{y}}_t \right)^2$ leads to parameter values that partly compensate for the error in $\boldsymbol{A}_t$, and some of these variables may apparently be far from the true values. The results in Case 6 also verify Theorem 2.

As shown in Table 2, Case 4, the BLUP algorithm results in small errors in $\widehat{u}_{\text{ref}}$, and correspondingly good mean trait predictions. As these predictions are not very sensitive to measurement errors (Cases 5 and 6) or errors in $\boldsymbol{A}_t$ (Case 6), the BLUP/PEM method appears to be a feasible solution for the microevolution–plasticity disentanglement problem introduced in Section 1, at least in cases with a single reference environment. The BLUP method is also more flexible than the GRAD method, in that additional fixed effects can be included in the model (Ch. 19, Walsh and Lynch, 2018).

The large errors in the estimated parameter values caused by measurement noise (Cases 2, 5 and 6) limits the usefulness of the parameter estimates for other purposes, as for example for finding heritabilities of reaction norm traits (Ch. 19, Walsh and Lynch, 2018). A similar situation occurs when the true parameter values vary over time (Table 4, Cases 8 and 9).

Tables 1, 2 and 4 include mean values and standard errors of the final value of the criterion function. Tests with a reduced model (Example 4, Case 10) show that $\sum \varepsilon_{t,final}^2$ increases for all realizations, although to various degree. This indicates that the value of $\sum \varepsilon_{t,final}^2$ may be used as a tool for model selection, but here a more detailed study is necessary.

The fact that predicted mean trait values $\widehat{\overline{a}}_t$ and $\widehat{\overline{\boldsymbol{b}}}_t$ may be quite good also with large errors in the estimated $\boldsymbol{G}$ and $\boldsymbol{P}$ parameter values, as discussed above and demonstrated in the simulations, raises difficult questions regarding model validation. In system identification in general, identified models are preferably

validated by use of data that have not been used for the identification (Ljung, 2010), and in an industrial context it is often possible to obtain such validation data. In the evolutionary context, however, the data series are normally obtained from cumbersome field observations over a small number of generations for a specific population of for example squirrels (Boutin and Lane, 2014). Independent data over the same number of generations for the same species may therefore be impossible to obtain. A possible validation approach could be to use a part of the data for identification and the rest for validation, but as seen in the simulations the dynamics in the responses are quite different in different time periods. A remaining possibility is some form of cross validation, but details of such an approach remain to be worked out.

Details regarding additional fixed effects in the dynamical BLUP model, various sources of drift, etc., also remain to be worked out. Finally, and most importantly, tests on real world data must also be left for further work.

Acknowledgments

References

Arnold, P. A., Kruuk, L. E., and Nicotra, A. B. How to analyse plant phenotypic plasticity in response to a changing climate. *New Phytologist*, 2019. 222(3):1235–1241. doi:10.1111/nph.15656.

Boutin, S. and Lane, J. E. Climate change and mammals: evolutionary versus plastic responses. *Evolutionary applications*, 2014. 7(1):29–41. doi:10.1111/eva.12121.

Bowers, E. K., Grindstaff, J. L., Soukup, S. S., Drilling, N. E., Eckerle, K. P., Sakaluk, S. K., and Thompson, C. F. Spring temperatures influence selection on breeding date and the potential for phenological mismatch in a migratory bird. *Ecology*, 2016. 97(10):2880–2891. doi:10.1002/ecy.1516.

Cantet, R., Angarita-Barajas, B., Forneris, N., and Munilla, S. Causal inference for the covariance between breeding values under identity disequilibrium. *Genetics, Selection, Evolution: GSE*, 2022. 54(1):64–64. doi:10.1186/s12711-022-00750-6.

Ergon, R. The environmental zero-point problem in evolutionary reaction norm modeling. *Ecology and Evolution*, 2018. 8(8):4031–4041. doi:10.1002/ece3.3929.

Ergon, R. Quantitative genetics state-space modeling of phenotypic plasticity and evolution. *Modeling, Identification and Control*, 2019. 40:51–69. doi:10.4173/mic.2019.1.5.

Ergon, R. The important choice of reference environment in microevolutionary climate response predictions. *Ecology and evolution*, 2022a. 12(4). doi:10.1002/ece3.8836.

Ergon, R. Microevolutionary system identification and climate response predictions. *Modeling, Identification and Control*, 2022b. 43(3):91–99. doi:10.4173/mic.2022.3.1.

Ergon, R. A BLUP derivation of the multivariate breeder's equation, with an elucidation of errors in blup variance estimates, and a prediction method for inbred populations. *Modeling, Identification and Control*, 2022c. 43(4):131–140. doi:10.4173/mic.2022.4.2.

Ergon, R. Dynamical BLUP modeling of reaction norm evolution, accommodating changing environments, overlapping generations, and multivariate data. *Ecology and Evolution*, 2023. 13(7):e10194. doi:10.1002/ece3.10194.

Ergon, T. and Ergon, R. When three traits make a line: Evolution of phenotypic plasticity and genetic assimilation through linear reaction norms in stochastic environments. *Journal of Evolutionary Biology*, 2017. 30(3):486–500. doi:10.1111/jeb.13003.

Goudet, J., Kay, T., and Weir, B. S. How to estimate kinship. *Molecular ecology*, 2018. 27(20):4121–4135. doi:10.1111/mec.14833.

Hadfield, J. D., Wilson, A. J., Garant, D., Sheldon, B. C., and Kruuk, L. E. The misuse of BLUP in ecology and evolution. *The American Naturalist*, 2010. 175(1):116–125. doi:10.1086/648604.

Henderson, C. R. Estimation of genetic parameters. *Ann. Math. Stat.*, 1950. 21:309–310.

klimaservicesenter, N. 2023. URL https://seklima.met.no/observations/.

Kruuk, L. E. Estimating genetic parameters in natural populations using the 'animal model'. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 2004. 359(1446):873–890. doi:10.1098/rstb.2003.1437.

Lande, R. Quantitative genetic analysis of multivariate evolution, applied to brain: body size allometry. *Evolution*, 1979. pages 402–416. doi:10.2307/2407630.

Lande, R. Adaptation to an extraordinary environment by evolution of phenotypic plasticity and genetic assimilation. *Journal of evolutionary biology*, 2009. 22(7):1435–1446. doi:10.1111/j.1420-9101.2009.01754.x.

Liu, W. *Introduction to hybrid vehicle system modeling and control, Appendix A: System Identification, State and Parameter Estimation Techniques*. John Wiley & Sons, 2013. doi:10.1002/9781119278924.app1.

Ljung, L. *System Identification: Theory for the User*. 2nd Edition, Prentice-Hall, Upper Saddle River, NJ, 1999.

Ljung, L. Perspectives on system identification. *Annual Reviews in Control*, 2010. 34(1):1–12. doi:10.1016/j.arcontrol.2009.12.001.

Ljung, L., Hjalmarsson, H., and Ohlsson, H. Four encounters with system identification. *European Journal of Control*, 2011. 17(5-6):449–471. doi:10.3166/ejc.17.449-471.

Lynch, M. and Walsh, B. Genetics and analysis of quantitative traits. sinauer associates. *Inc., Sunderland, MA*, 1998. 980.

Mathew, B., Léon, J., and Sillanpää, M. J. Impact of residual covariance structures on genomic prediction ability in multi-environment trials. *Plos one*, 2018. 13(7):e0201181. doi:10.1371/journal.pone.0201181.

Merilä, J. and Hendry, A. P. Climate change, adaptation, and phenotypic plasticity: the problem and the evidence. *Evolutionary applications*, 2014. 7(1):1–14. doi:10.1111/eva.12137.

Nussey, D. H., Wilson, A. J., and Brommer, J. E. The evolutionary ecology of individual phenotypic plasticity in wild populations. *Journal of Evolutionary Biology*, 2007. 20(3):831–844. doi:10.1111/j.1420-9101.2007.01300.x.

Price, G. R. et al. Selection and covariance. *Nature*, 1970. 227:520–521.

Robertson, A. A mathematical model of the culling process in dairy cattle. *Animal Science*, 1966. 8(1):95–108.

Robinson, G. K. That BLUP is a good thing: the estimation of random effects. *Statistical science*, 1991. pages 15–32.

Scheiner, S. M., Barfield, M., and Holt, R. D. The genetics of phenotypic plasticity. XVII. response to climate change. *Evolutionary Applications*, 2020. 13(2):388–399. doi:10.1111/eva.12876.

Stoica, P. and Söderström, T. *System identification*. Prentice-Hall International, 1989.

Thomson, C. E., Winney, I. S., Salles, O. C., and Pujol, B. A guide to using a multiple-matrix animal model to disentangle genetic and nongenetic causes of phenotypic variance. *PLoS One*, 2018. 13(10):e0197720. doi:10.1371/journal.pone.0197720.

Walsh, B. and Lynch, M. *Evolution and selection of quantitative traits*. Oxford University Press, 2018.

A. Proof of Theorem 3

In order to prove that the GRAD system given by Equations (2a) and (5a) is parameter identifiable (Ch. 6, Stoica and Söderström, 1989), we must show that there exists a unique set of values in $\boldsymbol{u}_{\mathrm{ref}}$, $\overline{a}_1$, $\overline{\boldsymbol{b}}_1$, $\boldsymbol{G}$ and $\boldsymbol{P}$, that minimizes the criterion function $\frac{1}{T}\sum_{t=1}^{T}\left(\overline{y}_t - \widehat{\overline{y}}_t\right)^2$ when $N \to \infty$. This implies that the system is parameter identifiable if and only if an error in one of the elements in these vectors and matrices, cannot be compensated by errors in other component, in a way that is valid for all generations from t to T.

From Equation (5a) follows that $\Delta\overline{a}_t$ and $\Delta\overline{\boldsymbol{b}}_t$ are determined by input-output and fitness information at time t. Since $\boldsymbol{u}_{t+1}'^T$ is unknown at time t, it thus follows from Equation (4), $\Delta\overline{y}_t = \Delta\overline{a}_t + \boldsymbol{u}_{t+1}'^T\Delta\overline{\boldsymbol{b}}_t + \Delta\boldsymbol{u}_t'^T\overline{\boldsymbol{b}}_t$, that an error in $\Delta\overline{a}_t$ cannot be compensated by an error in $\Delta\overline{\boldsymbol{b}}_t$, or vice versa. It thus suffices to prove that only perfect parameter values can give $\Delta\widehat{\overline{a}}_t \equiv \Delta\overline{a}_t$ according to Equation (5a).

From Equation (5a) follows that $\Delta\widehat{\overline{a}}_t/\mathrm{cov}\left(w_{i,t}, y_{i,t}\right) = \left(\widehat{G}_{aa} + \widehat{\boldsymbol{G}}_{ab}\boldsymbol{u}'_t\right)/\widehat{P}_{yy,t} = \widehat{N}_t/\widehat{P}_{yy,t}$, where $\mathrm{cov}\left(w_{i,t}, y_{i,t}\right)$ is known. After setting $\widehat{G}_{aa} = G_{aa}$, and the elements in the $1 \times q$ matrix $\widehat{\boldsymbol{G}}_{ab}$ and the $q \times q$ matrix $\widehat{\boldsymbol{G}}_{bb}$ to $\widehat{G}_{ab_j}$ and $\widehat{G}_{kj}$, respectively, we find $\widehat{N}_t =$

$G_{aa} + \sum_{j=1}^{q} \widehat{G}_{ab_j} u'_{j,t}$ and from Equation (5b) $\widehat{P}_{yy,t} =$

$$\begin{bmatrix} G_{aa} + 2\sum_{j=1}^{q} \widehat{G}_{ab_j} u'_{j,t} + \sum_{j=1}^{q} \widehat{G}_{1j} u'_{1,t} u'_{j,t} \\ + \sum_{j=1}^{q} \widehat{G}_{2j} u'_{2,t} u'_{j,t} + \cdots + \sum_{j=1}^{q} \widehat{G}_{qj} u'_{q,t} u'_{j,t} \\ + \widehat{\sigma}_v^2 + \sum_{j=1}^{q} u'^2_{j,t} \widehat{\sigma}_{\eta_j}^2 \end{bmatrix}.$$

The expression for $\widehat{N}_t/\widehat{P}_{yy,t}$ should be compared with the corresponding expression for $N_t/P_{yy,t}$ with true parameter values, and with perfect parameter estimates we then find $\widehat{N}_t/\widehat{P}_{yy,t} \equiv N_t/P_{yy,t}$, i.e., $\Delta\widehat{\overline{a}}_t \equiv \Delta\overline{a}_t$, and thus $\widehat{\overline{a}}_t \equiv \overline{a}_t$, provided that the initial value is $\widehat{\overline{a}}_1 \equiv \overline{a}_1$. It thus remains to prove that non-perfect parameter values cannot give $\widehat{N}_t/\widehat{P}_{yy,t} \equiv N_t/P_{yy,t}$, and that $\widehat{\overline{a}}_1 \neq \overline{a}_1$ cannot result in $\widehat{\overline{a}}_t \equiv \overline{a}_t$.

All unknown terms in $\widehat{N}_t$ and $\widehat{P}_{yy,t}$ have different input weighting factors $u'_{j,t}$, $u'_{k,t}u'_{j,t}$ or $u'^2_{j,t}$, where k and j varies from 1 to q, and where $u'_{j,t} = u_{j,t} - u_{j,ref}$. At any given time, an error in the reference environment $u_{j,ref}$ for an input variable $u_{j,t}$, can conceivably be compensated by an error in the reference environment for another input variable, or errors in parameter values, or possibly a combinations of such compensating errors. But since $u_{j,t}$ varies with time, the effect of an error in $u_{j,ref}$ also varies with time, and an error compensation that works at one generation cannot therefore work at all generations. In a similar way an error in a parameter value $\widehat{G}_{ab_j}$ or $\widehat{G}_{kj}$, can conceivably be compensated by a combination of errors in reference

environments or in other parameter values, but not in a way that works at all time steps with different values of $u'_{j,t}$, $u'_{k,t}u'_{j,t}$ and $u'^2_{j,t}$.

With persistent excitation of sufficient order (for example white noise as part of an input variable), and sufficiently long data, an error in a reference environment $u_{j,ref}$, or in a parameter value $\widehat{G}_{ab_j}$ or $\widehat{G}_{kj}$, cannot therefore, in a way that works for all generations t in a data series, be compensated by errors in other reference environments or other parameter values.

Finally, it is self-evident that $\widehat{\overline{a}}_1 \neq \overline{a}_1$ cannot result in $\widehat{\overline{a}}_t \equiv \overline{a}_t$ for all t from 1 to T, and that $\widehat{\overline{b}}_{j,1} \neq \overline{b}_{j,1}$ cannot result in $\widehat{\overline{b}}_{j,t} \equiv \overline{b}_{j,t}$ for all t from 1 to T. The system is therefore parameter identifiable.

B. GRAD results for small and large populations

In order to test the GRAD/PEM optimization method with population sizes $N = 100$ and $N = 10,000$, true data is here generated by means of the GRAD Equations (5a–c). Results are shown in Table A.1. Note the poor parameter estimates for $N = 100$, primarily caused by large errors in $\widehat{u}_{\text{ref}}$.

Table A.1: Estimation and prediction results by means of the GRAD algorithm, with true system responses generated by means of the GRAD model in Equations (5a–c). Results are based on 10 simulations with different realizations of all random input variables. The estimated reference value $\widehat{u}_{\text{ref}}$ are in all cases a free variable.
Case 11: Perfect values of $u_{1,t}$, $u_{2,t}$ and $y_{i,t}$, with $N = 100$.
Case 12: As Case 13, but with $N = 10,000$.
Case 13: As Case 12, but with measurement errors in $u_{1,t}$, $u_{2,t}$ and $y_{i,t}$.

Parameter	True value	Optimization results Case 11	Optimization results Case 12	Optimization results Case 13
$\widehat{G}_{b_1 b_1}$	1	1.09 ± 0.42	1.00 ± 0.00	1.01 ± 0.02
$\widehat{G}_{b_2 b_2}$	0.5	0.57 ± 0.30	0.50 ± 0.00	0.50 ± 0.02
$\widehat{G}_{ab_1}$	0	0.14 ± 0.65	0.01 ± 0.00	-0.02 ± 0.08
$\widehat{\sigma}_v^2$	5	5.59 ± 2.53	4.99 ± 0.01	$4,93 \pm 0.18$
$\sigma_{\eta_1}^2$	1	1.10 ± 0.44	1 ± 0.00	1.00 ± 0.04
$\sigma_{\eta_2}^2$	0.5	0.60 ± 0.52	0.50 ± 0.00	0.52 ± 0.05
$\widehat{\overline{b}}_{1,init}$	–	-5.10 ± 0.07	-5.13 ± 0.09	-5.09 ± 0.08
$\widehat{\overline{b}}_{2,init}$	–	-0.05 ± 0.03	-0.04 ± 0.04	-0.05 ± 0.03
$\widehat{u}_{\text{ref}}$	0	-0.07 ± 0.29	-0.06 ± 0.05	-0.07 ± 0.10
$\sum \varepsilon_{t,final}^2$	–	$(7 \pm 42)10^{-4}$	$(12 \pm 1)10^{-8}$	$(18 \pm 6)10^{-5}$
$\Delta_{30}^{\text{error}}\widehat{\overline{a}}_t \%$	–	-3 ± 15	0.34 ± 0.14	-3.03 ± 5.04
$\Delta_{30}^{\text{error}}\widehat{\overline{b}}_{1,t} \%$	–	0 ± 1	0.00 ± 0.00	0.27 ± 0.57
$\Delta_{30}^{\text{error}}\widehat{\overline{b}}_{2,t} \%$	–	0 ± 1	0.00 ± 0.00	0.23 ± 0.86

C. MATLAB code for Examples 2, 3 and 4

```
clear

tic

x=30;
T=60;
N=100;
wsquare=1000;
varz=200;
varu=0.5;
rho=0.25;
Gaa=5;
varv=1*Gaa;
Gab=0;
Gbb1=1;
Gbb2=0.5;
Gbb=[Gbb1 0 ; 0 Gbb2];
vareta1=1*Gbb1;
vareta2=1*Gbb2;
Paa=Gaa+varv;
Pab=Gab;
Pbb1=Gbb1+vareta1;
Pbb2=Gbb2+vareta2;
G=[Gaa Gab 0 ; Gab Gbb1 0 ; 0 0 Gbb2];
P=[Paa Pab 0 ; Pab Pbb1 0 ; 0 0 Pbb2];

for m=1:10
    m

%% Matrix in second part of relationship matrix
for i=1:N
    for j=1:N
        if i==j dA0(i,j)=0; end
        if i==j+1 dA0(i,j)=0.5-0.25; end
        if i==j-1 dA0(i,j)=0.5-0.25; end
    end
end
dA=dA0+0.25*ones(N,N)-0.25*eye(N);

%% Generate u and theta sequences
uplot=zeros(1,T);
du=zeros(1,T);
dtheta=zeros(1,T);
for i=2:T
    du(i)=sqrt(varu)*randn;
    dtheta(i)=du(i)*rho*sqrt(varz/(varu))+sqrt(varz*(1-rho^2))*randn;
    if i>10
        uplot(i)=(i-10)/20;
    end
    du2(i)=du(i-1);
end
u10=uplot+du;
```

```matlab
u20=uplot+du2;
theta=-20*uplot-dtheta;

%% Individual population traits around abar, bbar and cbar
a0=zeros(N,T);
b10=zeros(N,T);
b20=zeros(N,T);
a=zeros(N,T);
b1=zeros(N,T);
b2=zeros(N,T);
for t=1:T
    a0(:,t)=sqrt(Gaa)*randn(N,1);
    b10(:,t)=sqrt(Gbb1)*randn(N,1);
    b20(:,t)=sqrt(Gbb2)*randn(N,1);
    v(:,t)=sqrt(varv)*randn(N,1);
    eta1(:,t)=sqrt(vareta1)*randn(N,1);
    eta2(:,t)=sqrt(vareta2)*randn(N,1);
    A=eye(N);
    if t>30 A=eye(N)+0.25*dA;
        if t>40 A=eye(N)+0.5*dA;
            if t>50 A=eye(N)+1*dA;
            end
        end
    end
    a(:,t)=sqrtm(A)*a0(:,t);
    b1(:,t)=sqrtm(A)*b10(:,t);
    b2(:,t)=sqrtm(A)*b20(:,t);
    a(:,t)=a(:,t)-mean(a(:,t));
    b1(:,t)=b1(:,t)-mean(b1(:,t));
    b2(:,t)=b2(:,t)-mean(b2(:,t));
    v(:,t)=v(:,t)-mean(v(:,t));
    eta1(:,t)=eta1(:,t)-mean(eta1(:,t));
    eta2(:,t)=eta2(:,t)-mean(eta2(:,t));
end

%% Simulation of true system
bbar10=-5*ones(1,T+1);
bbar20=0*ones(1,T+1);
ybar0=-0*ones(1,T);
abar0=zeros(1,T+1);
for t=1:T
%   Gaat(t)=Gaa+0.08333333*(t-1);                  % For Example 3
    abar0(1)=ybar0(1)-bbar10(1)*u10(1)-bbar20(1)*u20(1);
    bterm1=(bbar10(t)+b1(:,t)+eta1(:,t)).*u10(t);
    bterm2=(bbar20(t)+b2(:,t)+eta2(:,t)).*u20(t);
    Y0(:,t)=abar0(t)+a(:,t)+v(:,t)+bterm1+bterm2;
    W0(:,t)=10*exp(-(Y0(:,t)-theta(t)).^2/(2*wsquare));
    W0(:,t)=round(W0(:,t));
    Wbar0(t)=mean(W0(:,t));
    X=ones(N,1);
    U=[u10(t)*eye(N)  u20(t)*eye(N)];
    r=varv+u10(t)^2*vareta1+u20(t)^2*vareta2;
    A=eye(N);
    if t>30 A=eye(N)+0.25*dA;
```

```matlab
            if t>40 A=eye(N)+0.5*dA;
                if t>50 A=eye(N)+1*dA;
                end
            end
    end
    G11=Gaa*A;
%   G11=Gaat(t)*A;                              % For Example 3
    G12=Gab*A;
    G22=Gbb1*A;
    G33=Gbb2*A;
    Z=[eye(N) U];
    Gm=[G11 G12 0*eye(N); G12 G22 0*eye(N) ; 0*eye(N) 0*eye(N) G33];
    M=[X'*X X'*Z
        Z'*X Z'*Z+r*inv(Gm)];

    Ym=[ X'*Y0(:,t) ; Z'*Y0(:,t) ];
    effects=inv(M)*Ym;
    cova=cov(W0(:,t),effects(2:N+1));
    Dabar=((N-1)/N)*cova(1,2)/Wbar0(t);
    covb1=cov(W0(:,t),effects(N+2:2*N+1));
    Dbbar1=((N-1)/N)*covb1(1,2)/Wbar0(t);
    covb2=cov(W0(:,t),effects(2*N+2:3*N+1));
    Dbbar2=((N-1)/N)*covb2(1,2)/Wbar0(t);
    abar0(t+1)=abar0(t)+Dabar;
    bbar10(t+1)=bbar10(t)+Dbbar1;
    bbar20(t+1)=bbar20(t)+Dbbar2;
end
ybar0=mean(Y0);

%% Measurement errors
Y0=Y0+1*rand(N,T)-0.5;
ybar0=mean(Y0);
u10=u10+0.1*sqrt(varu)*randn;
u20=u20+0.1*sqrt(varu)*randn;

%% Short data
u1=u10(T-x+1:T);
u2=u20(T-x+1:T);
Y=Y0(:,T-x+1:T);
abar=abar0(T-x+1:T)-ybar0(T-x+1);
bbar1=bbar10(T-x+1:T);
bbar2=bbar20(T-x+1:T);
ybar=ybar0(T-x+1:T)-ybar0(T-x+1);
W=W0(:,T-x+1:T);
Wbar=mean(W);

deltaabar(m)=abar(30)-abar(1);

%% Contraints
Gbb1_min=0;Gbb1_max=10;
Gbb2_min=0;Gbb2_max=10;
% Gbb2_min=0.000001;Gbb2_max=0.000001;        % For Example 4
Gab_min=-10;Gab_max=10;
vare_min=0;vare_max=50;
```

```
varw1_min=0;varw1_max=10;
varw2_min=0;varw2_max=10;
% varw2_min=0.000001;varw2_max=0.000001;        % For Example 4
bbarinit1_min=-10; bbarinit1_max=0;
bbarinit2_min=-10; bbarinit2_max=10;
Gbbmin=[Gbb1_min,Gbb2_min];
Gbbmax=[Gbb1_max,Gbb2_max];
varwmin=[varw1_min,varw2_min];
varwmax=[varw1_max,varw2_max];
bbarinitmin=[bbarinit1_min bbarinit2_min];
bbarinitmax=[bbarinit1_max bbarinit2_max];
uref_min=-0.0000001;uref_max=0.0000001;
uref_min=-2;uref_max=2;
% uref_min=0.199999;uref_max=0.200001;
% uref_min=0.999999;uref_max=1.000001;           % For Table 3
% uref_min=-0.500001;uref_max=-0.499999;


par_lb=[Gbbmin,Gab_min,vare_min,varwmin,bbarinitmin,uref_min];
par_ub=[Gbbmax,Gab_max,vare_max,varwmax,bbarinitmax,uref_max];

par_guess=[zeros(1,7) 1 1];
par_known=[Gaa];
ybarhat_init=0;

%% fmincon:

% dA=zeros(N,N);                                 % For Case 6

Aineq=[]; Bineq=[]; Aeq=[]; Beq=[];

fun_objective_handle_Example_2=...
    @(par)fun_objective_Example_2(par,par_known,u1,u2,Y,ybar,dA,W,Wbar,
        ybarhat_init,N,T,x);

fun_constraints_handle_Example_2=...
    @(par)fun_constraints_Example_2(par,par_known,u1,u2,Y,ybar,dA,W,Wbar,
        ybarhat_init,N,T,x);

[par_opt,fval,exitflag,output,lambda,grad,hessian] =...
fmincon(fun_objective_handle_Example_2,par_guess,Aineq,Bineq,Aeq,Beq,par_lb,
    par_ub,fun_constraints_handle_Example_2);

output
fval
Fval(m)=fval;
opt_results(m,:)=par_opt;

%% Simulation with adapted model
Gbb1opt=par_opt(1);
Gbb2opt=par_opt(2);
Gabopt=par_opt(3);
vareopt=par_opt(4);
varw1opt=par_opt(5);
varw2opt=par_opt(6);
```

```
bbar1opt=par_opt(7);
bbar2opt=par_opt(8);
urefopt=par_opt(9);
u1opt=u1-urefopt;
u2opt=u2-urefopt;
bbarhat1=bbar1opt*ones(x,1);
bbarhat2=bbar2opt*ones(x,1);
ybarhat=0*ones(x,1);

% Gbb1opt=Gbb1;                                    % For use in Table 4, Case 7
% Gbb2opt=Gbb2;
% Gabopt=Gab;
% vareopt=Gaa;
% varw1opt=Gbb1;
% varw2opt=Gbb2;
% bbar1opt=bbar10(31);
% bbar2opt=bbar20(31);
% bbarhat1=bbar1opt*ones(x,1);
% bbarhat2=bbar2opt*ones(x,1);
% u1opt=u1;
% u2opt=u2;

for t=1:x-1
    abarhat(1)=ybar(1)-bbarhat1(1)*u1opt(1)-bbarhat2(1)*u2opt(1);
    X=ones(N,1);
    U=[u1opt(t)*eye(N)  u2opt(t)*eye(N)];
    r=vareopt+u1opt(t)^2*varw1opt+u2opt(t)^2*varw2opt;
    A=eye(N);
    if t>0 A=eye(N)+0.25*dA;
        if t>10 A=eye(N)+0.5*dA;
            if t>20 A=eye(N)+1*dA;
            end
        end
    end
    G11=Gaa*A;
    G12=Gabopt*A;
    G22=Gbb1opt*A;
    G33=Gbb2opt*A;
    Z=[eye(N) U];
    Gm=[G11 G12 0*eye(N); G12 G22 0*eye(N) ; 0*eye(N) 0*eye(N) G33];
    M=[X'*X  X'*Z
        Z'*X  Z'*Z+r*inv(Gm)];
    Ym=[ X'*Y(:,t) ; Z'*Y(:,t) ];
    effects=inv(M)*Ym;
    cova=cov(W(:,t),effects(2:N+1));
    Dabarhat=((N-1)/N)*cova(1,2)/Wbar(t);
    covb1=cov(W(:,t),effects(N+2:2*N+1));
    Dbbarhat1=((N-1)/N)*covb1(1,2)/Wbar(t);
    covb2=cov(W(:,t),effects(2*N+2:3*N+1));
    Dbbarhat2=((N-1)/N)*covb2(1,2)/Wbar(t);
    abarhat(t+1)=abarhat(t)+Dabarhat;
    bbarhat1(t+1)=bbarhat1(t)+Dbbarhat1;
    bbarhat2(t+1)=bbarhat2(t)+Dbbarhat2;
```

```matlab
        ybarhat(t+1)=abarhat(t+1)+bbarhat1(t+1)*u1opt(t+1)+bbarhat2(t
            +1)*u2opt(t
            +1);
end

%% GRAD predictions
bbarhat1G=bbar1opt*ones(x,1);
bbarhat2G=bbar2opt*ones(x,1);
ybarhatG=-0*ones(x,1);

for t=1:x-1
    abarhatG(1)=ybar(1)-bbarhat1G(1)*u1opt(1)-bbarhat2G(1)*u2opt(1);
    Paaopt=Gaa+vareopt;
    Pbb1opt=Gbb1opt+varw1opt;
    Pbb2opt=Gbb2opt+varw2opt;
    Pyy=Paaopt+2*Gabopt*u1opt(t)+u1opt(t)^2*Pbb1opt+u2opt(t)^2*Pbb2opt;
    covWy=(N-1)*cov(W(:,t),Y(:,t))/N;
    betay=inv(Pyy)*covWy(1,2)/Wbar(t);
    abarhatG(t+1)=abarhatG(t)+(Gaa+Gabopt*u1opt(t))*betay;
    bbarhat1G(t+1)=bbarhat1G(t)+(Gabopt+Gbb1opt*u1opt(t))*betay;
    bbarhat2G(t+1)=bbarhat2G(t)+(0+Gbb2opt*u2opt(t))*betay;
end

%% Results
Totalabar=abar(1)-abar(x);
Totalabarhat=abarhat(1)-abarhat(x);
Rel_total_abar_error(m)=(Totalabarhat-Totalabar)/Totalabar;
Totalbbar1=bbar1(1)-bbar1(x);
Totalbbarhat1=bbarhat1(1)-bbarhat1(x);
Rel_total_bbar1_error(m)=(Totalbbarhat1-Totalbbar1)/Totalbbar1;
Totalbbar2=bbar2(1)-bbar2(x);
Totalbbarhat2=bbarhat2(1)-bbarhat2(x);
Rel_total_bbar2_error(m)=(Totalbbarhat2-Totalbbar2)/Totalbbar2;

Totalabar=abar(1)-abar(x);
TotalabarhatG=abarhatG(1)-abarhatG(x);
Rel_total_abar_error_G(m)=(TotalabarhatG-Totalabar)/Totalabar;
Totalbbar1G=bbar1(1)-bbar1(x);
Totalbbarhat1G=bbarhat1G(1)-bbarhat1G(x);
Rel_total_bbar1_error_G(m)=(Totalbbarhat1G-Totalbbar1)/Totalbbar1;
Totalbbar2=bbar2(1)-bbar2(x);
Totalbbarhat2G=bbarhat2G(1)-bbarhat2G(x);
Rel_total_bbar2_error_G(m)=(Totalbbarhat2G-Totalbbar2)/Totalbbar2;
end

%% Plots
ybar=[NaN*ones(1,T-x) ybar];
ybarhat=[NaN*ones(1,T-x) ybarhat'];
abar=[NaN*ones(1,T-x) abar];
abarhat=[NaN*ones(1,T-x) abarhat];
abarhatG=[NaN*ones(1,T-x) abarhatG];
bbar1=[NaN*ones(1,T-x) bbar1];
bbarhat1=[NaN*ones(1,T-x) bbarhat1'];
bbarhat1G=[NaN*ones(1,T-x) bbarhat1G'];
bbar2=[NaN*ones(1,T-x) bbar2];
```

```matlab
bbarhat2=[NaN*ones(1,T-x) bbarhat2'];
bbarhat2G=[NaN*ones(1,T-x) bbarhat2G'];

figure(2)
subplot(3,2,1)                              % (3,2,2) for Panel d)
plot(ybar,'b'), hold on
plot(ybarhat,'.b')
plot(abar,'g','LineWidth',2)
plot(abarhat,'.k','LineWidth',2)
plot(abarhatG,':m','LineWidth',2)
title('   Mean phenotype and intercept')
ylabel('days')                              % Don't use for Panel d)
axis([T-x T -20 10]), hold off, grid
text(T-x-1.5,15,'a)','FontSize',14)         % d) for Panel d)

subplot(3,2,3)                              % (3,2,4) for Panel e)
plot(bbar1,'g','LineWidth',2), hold on
plot(bbarhat1,'.k','LineWidth',2)
plot(bbarhat1G,':m','LineWidth',2)
hold off, grid
title('Mean plasticity slope one')
ylabel('days/^oC')                          % Don't use for Panel e)
axis([T-x T -6.5 -4.5])
text(T-x-1.5,-4.16,'b','FontSize',14)       % e) for Panel e)

subplot(3,2,5)                              % (3,2,6) for Panel f)
plot(bbar2,'g','LineWidth',2), hold on
plot(bbarhat2,'.k','LineWidth',2)
plot(bbarhat2G,':m','LineWidth',2)
hold off, grid
xlabel('Generation [t]')
title('Mean plasticity slope two')
ylabel('days/^oC')                          % Don't use for Panel f)
axis([T-x T -1 0.5])
text(T-x-1.5,0.75,'c)','FontSize',14)       % f) for Panel f)

%% Total results
Mean=mean(opt_results)
Std=std(opt_results)

Mean_abar_error=mean(Rel_total_abar_error);
Std_abar_error=std(Rel_total_abar_error);
abar_results=[Mean_abar_error Std_abar_error]
Mean_bbar1_error=mean(Rel_total_bbar1_error);
Std_bbar1_error=std(Rel_total_bbar1_error);
bbar1_results=[Mean_bbar1_error Std_bbar1_error]
Mean_bbar2_error=mean(Rel_total_bbar2_error);
Std_bbar2_error=std(Rel_total_bbar2_error);
bbar2_results=[Mean_bbar2_error Std_bbar2_error]

Mean_abar_error_G=mean(Rel_total_abar_error_G);
Std_abar_error_G=std(Rel_total_abar_error_G);
abar_results_G=[Mean_abar_error_G Std_abar_error_G]
Mean_bbar1_error_G=mean(Rel_total_bbar1_error_G);
```

```
Std_bbar1_error_G=std(Rel_total_bbar1_error_G);
bbar1_results_G=[Mean_bbar1_error_G Std_bbar1_error_G]
Mean_bbar2_error_G=mean(Rel_total_bbar2_error_G);
Std_bbar2_error_G=std(Rel_total_bbar2_error_G);
bbar2_results_G=[Mean_bbar2_error_G Std_bbar2_error_G]
Mean_fval=mean(Fval);
Std_fval=std(Fval);
fval_results=[Mean_fval Std_fval]

toc

function f = fun_objective_Example_2(par,par_known,u1,u2,Y,ybar,dA,W,Wbar,
    ybarhat_init,N,T,x)

Gbb1e=par(1);
Gbb2e=par(2);
Gabe=par(3);
varee=par(4);
varw1e=par(5);
varw2e=par(6);
bbarhat1_init=par(7);
bbarhat2_init=par(8);
uref=par(9);
u1e=u1-uref;
u2e=u2-uref;
Gaa=par_known(1);
u1e=u1;
u2e=u2;
bbarhat1=bbarhat1_init*ones(x+1,1);
bbarhat2=bbarhat2_init*ones(x+1,1);
ybarhat=ybarhat_init*ones(x+1,1);

for t=1:x-1
    abarhat(1)=ybar(1)-bbarhat1(1)*u1e(1)-bbarhat2(1)*u2e(1);
    X=ones(N,1);
    U=[u1e(t)*eye(N) u2e(t)*eye(N)];
    r=varee+u1e(t)^2*varw1e+u2e(t)^2*varw2e;
    A=eye(N);
    if t>0 A=eye(N)+0.25*dA;
        if t>10 A=eye(N)+0.5*dA;
            if t>20 A=eye(N)+1*dA;
            end
        end
    end
    G11=Gaa*A;
    G12=Gabe*A;
    G22=Gbb1e*A;
    G33=Gbb2e*A;
    Z=[eye(N) U];
    Gm=[G11 G12 0*eye(N) ; G12 G22 0*eye(N) ; 0*eye(N) 0*eye(N) G33];
    M=[X'*X X'*Z
        Z'*X Z'*Z+r*pinv(Gm)];
    Ym=[ X'*Y(:,t) ; Z'*Y(:,t) ];
    effects=inv(M)*Ym;
```

```matlab
        beta(t+1)=effects(1);
        cova=cov(W(:,t),effects(2:N+1));
        Dabarhat=((N-1)/N)*cova(1,2)/Wbar(t);
        covb1=cov(W(:,t),effects(N+2:2*N+1));
        Dbbarhat1=((N-1)/N)*covb1(1,2)/Wbar(t);
        covb2=cov(W(:,t),effects(2*N+2:3*N+1));
        Dbbarhat2=((N-1)/N)*covb2(1,2)/Wbar(t);
        abarhat(t+1)=abarhat(t)+Dabarhat;
        bbarhat1(t+1)=bbarhat1(t)+Dbbarhat1;
        bbarhat2(t+1)=bbarhat2(t)+Dbbarhat2;
        ybarhat(t+1)=abarhat(t+1)+bbarhat1(t+1)*u1e(t+1)+bbarhat2(t+1)*u2e(t+1);
end
     f=sum((( ybar(1:x)'-ybarhat(1:x)).^2)/x);
end

function [cineq,ceq]=fun_constraints_Example_2(par,par_known,u1,u2,Y,ybar,dA
    ,W,Wbar,ybarhat_init,N,T,x)
cineq =   [];
ceq = [];
end
```