SUPPLEMENTARY INFORMATION

Evolutionary design of regulatory control. II. Robust error-correcting feedback increases genetic and phenotypic variability

Steven A. Frank*

1 Computer simulations methods
1.1 Basic population genetics
1.2 Encoding of control loop for phenotype
2 Simulation experiments
3 Results
3.1 Mean and standard deviation in fitness
3.2 Genetic and phenotypic variability
References

1 Computer simulations methods

I wrote computer simulations in C++ to study the evolutionary dynamics of feedback control systems. Table 1 shows the parameters of the simulation.

The C++ source code is available at https://doi.org/10.5281/zenodo.1405968. The Mathematica code used to analyze the simulation data and produce the figures and the raw simulation data are available at https://doi.org/10.5281/zenodo.1405960. I generated 64 bit random numbers with the pcg64 routine of the PCG random number generator, described at http://www.pcg-random.org/.

1.1 Basic population genetics

The evolutionary aspects follow standard conventions for evolving populations. Each individual in the population has one (haploid) copy of each chromosome. All individuals have one primary chromosome that defines the inherited genetic determination of various traits that influence phenotype. In some simulations, individuals also have a secondary chromosome that defines a stochastic component for the expression of each trait, as described in the following subsection.

A chromosome encodes a linear sequence of genes, each gene (locus) describing an attribute that influences the individual’s characteristics. Each gene may have alternative (allelic) values.

A single generation in the standard life cycle begins by assigning fitness values to each individual of the population based on the allelic values of its chromosomes (see main text). The current parental population is then used to make a new descendant population of progeny.

Each progeny is made by selecting two parents, each parent chosen with a probability in proportion to its fitness. Each progeny chromosome is made by recombining the matching chromosomes from the two parents. In particular, the first gene locus for the progeny is chosen by selecting the chromosome from one parent if a uniform random number on [0, 1] (U01) is less than one half, otherwise selecting the chromosome from the other parent.

If individuals have two chromosomes, then each chromosome is built in parallel, by taking the same parent as the source for each locus of each chromosome. Thus, if the progeny has two chromosomes, its alleles at the first locus of each chromosome come from the same parent, its alleles at the second locus of each chromosome come from the same parent, and so on. I used this nonstandard approach in order to make the loci that determine traits on the first chromosome be cotransmitted with the matching loci that determine stochasticity of expression on the second chromosome, with no recombination between matched pairs.

Moving along a linear chromosome, the next gene

* Department of Ecology and Evolutionary Biology, University of California, Irvine, CA 92697–2525, USA
web: https://stevefrank.org
locus is chosen from the other parent if a new U01 is less than \( \text{rec} \), the recombination rate, otherwise the next locus is taken from the same parent. That process repeats along the length of the chromosomes, forming the new progeny chromosomes.

Each progeny allelic value is mutated with probability \( \text{mut} \). If mutated, then a random uniform number on \([-\text{mutStep}, \text{mutStep}]\) is added to the current allelic value. On the chromosome that controls stochastic expression of traits (see below), allelic values less than zero are set to zero.

The process of choosing parents (with replacement) and making progeny chromosomes repeats until the progeny population size matches the target number, \( \text{popSize} \). I used the alias method algorithm for repeatedly choosing parents with replacement in proportion to their fitnesses, see https://en.wikipedia.org/w/index.php?title=Alias_method&oldid=847648253.

I validated this C++ code with respect to basic population genetics by comparing simulation results with a single chromosome and single locus to standard analytic theory for mutation-selection balance (Frank & Slatkin, 1990).

### 1.2 Encoding of control loop for phenotype

The primary chromosome encodes the inherited genetic component for the sequence of controller parameters, \( p_1, p_2, q_0, q_1, q_2 \), with \( p_0 \equiv 0 \) and not encoded as a genetic locus. If the loop has double feedback, then two additional loci encode \( r \) and \( k \).

If \( \text{stochWt} \) is greater than zero then, in each individual, the actual controller parameters include a stochastic component of expression. The amount of stochasticity for each parameter is determined by a matching locus on the secondary chromosome.

For example, the value of \( p_1 \) in a particular individual is the value of its first locus on the primary chromosome multiplied by \( 2^z \), in which \( z \) is a random number drawn from a Gaussian distribution with mean of zero and standard deviation of \( \text{stochWt} \times \delta_1 \), with \( \delta_1 \) as the value of the first locus on the secondary chromosome.

This calculation for each locus sets all of the controller parameters. The parameter \( \text{loop} \) specifies whether the control loop is open, single, or double.

The parameter \( \text{aSD} \) determines the stochastic fluctuations in the plant process parameter, \( \alpha \). In particular, the actual value of \( \alpha \) in any individual is \( \alpha = \sqrt{1 + \gamma} \times 2^\xi \), in which \( \xi \) is a random number drawn from a Gaussian distribution with mean of zero and standard deviation of \( \text{aSD} \).

## 2 Simulation experiments

I ran three simulation experiments with factorial combinations of parameters. Default values from Table 1 were used unless otherwise noted. In all simulations, the three alternative control loop designs of open, single, and double were also varied factorially in relation to the other parameters. In the analyses and figures, the open, single, and double loop designs were encoded as \{O, S, D\}, respectively.

The goal of the simulations is to evaluate sensitivity around the optimum. Thus, in each run, I initialized all individuals with an optimum genotype, and
then mutated each genotype using the standard mutation process for that run.

In Experiment A, \( mut = 10^{-2} \) and varied parameter values were \( gen = \{8 \times 10^2, 1.6 \times 10^3, 3.2 \times 10^3\}; aSD = \{0, 0.25, 0.5\}; \) stochWt = \{0, 0.25, 0.5\}.

In Experiment B, varied parameter values were \( mut = \{10^{-4}, 10^{-3}, 10^{-2}\}; rec = \{0, 0.25, 0.5\}; fitVar = \{10^{-2}, 10^{-1}, 10^0\}. \)

In Experiment C, varied parameter values were \( popSize = \{2 \times 10^3, 8 \times 10^3, 3.2 \times 10^4\}; mut = \{10^{-4}, 10^{-3}, 10^{-2}\}; fitVar = \{10^{-2}, 10^{-1}, 10^0\}. \)

In Experiment D, \( popSize = 3.2 \times 10^4 \) and varied parameter values were \( mut = \{10^{-4}, 10^{-3}, 10^{-2}\}; fitVar = \{10^{-2}, 10^{-1}, 10^0\}. \) This experiment ran three replicates for each parameter combination.

### 3 Results

#### 3.1 Mean and standard deviation in fitness

How does the robustness of error-correcting feedback influence fitness? To study that question, I analyzed the mean and standard deviation in fitness response in relation to variations in the experimental parameters.

For each experiment, I first evaluated which of the varying parameters explained a significant fraction of the observed variance in response, using standard ANOVA analysis (see supplemental Mathematica file for ANOVAs and additional analyses). In Experiment A, the main effects and interaction of the parameters loop and aSD explained 90% of the observed variance in mean fitness and 89% of the observed variance in the standard deviation of fitness.

Based on these results, Figure 5 of the main text
shows the mean and standard deviation responses as a function of the loop and aSD parameters, averaged over the other nonsignificant parameter values. I used the same approach of averaging over the nonsignificant parameter values for all experiments.

In Experiment B, the ANOVA model loop + mut + fitVar + loop \times fitVar explains 69% of the variance in mean fitness and 76% of the variance in the standard deviation of fitness. Figure 6 of the main text shows the fitness response as a function of these model parameters.

In Experiment C, the same ANOVA model as in Experiment B explained similar amounts of the total variances, and had similar response plots for mean and standard deviation in fitness (not shown). Experiment C showed that popSize had little effect on fitness over the range of population sizes analyzed.

### 3.2 Genetic and phenotypic variability

An individual expresses several controller parameters. Each parameter is determined by a combination of an inherited central value and an inherited tendency for stochastic variability around the central value (see section 1.2). I refer to these two components as genetic and phenotypic variability, respectively, noting that the tendency for phenotypic variability is an inherited trait.

In the final generation of each simulation run, I measured, for each individual in the population, the inherited genetic central value and the inherited tendency for stochasticity for each controller parameter. Thus, each parameter has two associated probability distributions that describe the genetic and phenotypic components of variability in the population. The architecture of control for homeostasis and step response influences the accumulation of variability. The theory in this article makes the key prediction that more robust control architectures will tend to accumulate more variability.

Various simulation parameters that determine evolutionary dynamics also influence the distributions that describe variability. For example, one would expect that higher mutation rates would typically lead to more variability.

Figure S1 summarizes the distributions of variability for Experiment A. In that experiment, the number of generations, gen, and the intensity of perturbation to the plant parameter, aSD, had relatively little influence on the patterns of variability (not shown). The figure summarizes the results for the third variable parameter of the simulations, stochWt, and for the three different loop architectures of open, single feedback, and double feedback.

In Fig. S1, the top row shows the distributions of inherited central values for traits. The bottom row shows the distributions of inherited tendencies for stochasticity. The columns from left to right show the values for the controller parameters p1, p2, q0, q1, and q2.

In each panel, a single vertical set of lines and dots describes a probability distribution. The lower dot is the 1st percentile, the lower line is the 5–25th percentile range, the middle dot is the 50th percentile, the upper line is the 75–95th percentile range, and the upper dot is the 99th percentile. The number in the upper left corner of each panel shows the maximum value for that panel. The minimum is always zero.

The three sets of distributions in each panel show the three control architectures, which are, from left to right, open, single, and double loops. Within each set, the distributions show, from left to right, increasing values of stochWt, with values of \{0,0.25,0.5\}. In the bottom row, no distributions are shown when stochWt = 0, because there is no stochasticity for controller parameter expression when the stochastic weighting is zero.

The information in Fig. S1 provides a good summary of the raw data on allelic variability in the simulated populations. However, it is difficult to compare that raw summary with the key prediction that more robust control architectures will accumulate more variability. Figure 2 of the main text expresses that key prediction by showing that fitness falls off more slowly around the optimal parameter value as the control architecture becomes more robust.

We can transform the raw distribution data in Fig. S1 into cumulative distribution function (CDF) curves (Fig. 7 of the main text). The detailed methods are described by the Mathematica source code included as a supplemental file. I briefly summarize the methods here.

Figure S2 plots the data for stochWt = 0.25, which corresponds to the middle distribution of each set of three in the top row of Fig. S1 and the left distri-
bution in each set of two in the bottom row of that figure.

Each curve in Figure S2 matches a percentile distribution in Fig. S1, with percentiles shown as dots and lines as described above. The curves of Figure S2 are CDFs calculated by smoothing the underlying percentile data from the simulations, which were recorded as the 0, 1, 2, . . . , 100 percentiles. The different colors show the control loop architecture: open loop (blue), single feedback loop (gold), and double feedback loop (green).

The top row of Figure S2 shows the inherited central tendency of each controller parameter value, the inherited genetic value. I centered each CDF at its median value, $\theta^*$, and then plotted the x axis as $\theta^* \times 2^x$ for $x \in [-1/2, 1/2]$. Thus, the x axis is a logarithmic scaling that runs from $\theta^*/\sqrt{2}$ to $\sqrt{2} \theta^*$.

The smaller the slope of a CDF curve, the wider the distribution of values. All of the panels in the top row show increasing variability with an increase in error-correcting feedback, except the second panel. In the second panel, for parameter $p_2$, the single feedback loop (gold) is highly variable because its optimum is zero, and so small amounts of variability show up as a large logarithmic range around a median that is very close to zero.

The bottom row of Fig. S2 shows the CDFs for the stochastic tendency for variability. The horizontal axis shows values of $x \in [0, 1]$. The flatter the CDF curve, the greater the occurrence of high values for the stochastic tendency for variability. Once again, an increase in robustness via greater error-correcting feedback associates with an increase in the variability of the expressed controller parameters.

Figure 7 of the main text shows the same CDF plots for Experiment C. Among the variable parameters of that simulation experiment, the figure shows the largest population size, $popSize = 3.2 \times 10^4$, and middle values of $mut = 10^{-3}$ and $fitVar = 10^{-1}$. This experiment also shows a significant increase in variability as the control architecture becomes more robust.

References

Frank, S. A. & Slatkin, M. (1990). The distribution of allelic effects under mutation and selection. *Genetical Research*, 55, 111-117.