B_s^0-$\bar{B}_s^0$ Mixing Limits Note

Jamie E. Hegarty and Phil Gutierrez

November 7, 2004

Using an unbinned likelihood amplitude fitting method with a toy Monte Carlo simulation of B_s^0-$\bar{B}_s^0$ mixing events, we have determined preliminary sensitivity values for a measurement of dilution.

1 The Math

The lifetime distributions of the mixed (M) and unmixed (U) states are given by:

$$U(t) = e^{-\frac{t}{\tau}}[1 + \cos(\Delta m t)], \quad \text{and} \quad M(t) = e^{-\frac{t}{\tau}}[1 - \cos(\Delta m t)] \quad \Rightarrow \quad \frac{U(t) - M(t)}{U(t) + M(t)} = \cos(\Delta m t)$$

where τ is the mean lifetime of the B_s^0. With mistagging included, the distributions of mixed ($N_m(t)$) and unmixed ($N_u(t)$) B_s^0 become:

$$N_m(t) = (1-\alpha)M(t) + \alpha U(t) = (1-\alpha)\{e^{-\frac{t}{\tau}}[1 - \cos(\Delta m t)]\} + \alpha\{e^{-\frac{t}{\tau}}[1 + \cos(\Delta m t)]\}$$
$$N_u(t) = (1-\alpha)U(t) + \alpha M(t) = (1-\alpha)\{e^{-\frac{t}{\tau}}[1 + \cos(\Delta m t)]\} + \alpha\{e^{-\frac{t}{\tau}}[1 - \cos(\Delta m t)]\}$$

where α is the mistag rate, and dilution D is defined as $1-2\alpha$. Since $N_m(t)$ and $N_u(t)$ differ by only the sign of the $\cos(\Delta m t)$ terms, an extra paramter "`tag`" may be used, such that `tag` $=1$ corresponds to the mixed stated, and `tag` $=-1$ corresponds to the unmixed state. The separate $N_m(t)$ and $N_u(t)$ can be generalized to a single distribution function:

$$N(t, \mathtt{tag}) = (1-\alpha)\{e^{-\frac{t}{\tau}}[1 - \mathtt{tag}\cdot\cos(\Delta m t)]\} + \alpha\{e^{-\frac{t}{\tau}}[1 + \mathtt{tag}\cdot\cos(\Delta m t)]\} \tag{1}$$

However, Gaussian smearing to account for the time resolution of the detector must also be factored in. The time-smeared distributions of the mixed and unmixed B_s^0 are found by convoluting $N(t, \mathtt{tag})$ with the (normalized) Gaussian $G(t) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-(\frac{t}{\sqrt{2}\sigma})^2}$:

$$f(t, \mathtt{tag}) = \frac{(1-\alpha)}{\sqrt{2\pi}\,\sigma} \int_{-\infty}^{\infty} e^{-(\frac{t-t'}{\sqrt{2}\sigma})^2} e^{-\frac{t'}{\tau}} [1 - \mathtt{tag}\cdot\cos(\Delta m t')]\; dt\; \ldots$$
$$+ \frac{\alpha}{\sqrt{2\pi}\,\sigma} \int_{-\infty}^{\infty} e^{-(\frac{t-t'}{\sqrt{2}\sigma})^2} e^{-\frac{t'}{\tau}} [1 + \mathtt{tag}\cdot\cos(\Delta m t')]\; dt$$

Integrating and making a few simplifying substitutions yields a function suitable to fit with:

$$f(t, \mathtt{tag}) = (1-2\alpha)\frac{e^B}{2\tau}\left[1 + \mathrm{erf}(A) - \mathtt{tag}\cdot w_r^- \, e^{-A^2}\right] + \alpha\frac{e^B}{2\tau}\left[1 + \mathrm{erf}(A) + \mathtt{tag}\cdot w_r^- \, e^{-A^2}\right] \tag{2}$$

$$A = \frac{t}{\sqrt{2}\,\sigma} - \frac{\sigma}{\sqrt{2}\,\tau}\ ; \qquad B = \frac{\sigma^2}{2\tau^2} - \frac{t}{\tau}\ ; \qquad C = \frac{\sigma\Delta m}{\sqrt{2}}$$

where w_r^- is the real part of the complex error function $w(z)$ [2], evaluated at $z = C - iA$. It is important to note that as A increases, the term $w_r^- \, e^{-A^2} \to \cos(2AC)$.

2 Monte Carlo Simulation

B_s^0 events are generated with Root [1] using a toy Monte Carlo (MC) simulation, as follows:

1. For each event, the proper lifetime t_p of the B_s^0 is first selected from a perfect exponential distribution with a mean at $\tau = 1.5 \times 10^{-12}s$. Next, a random time "smearing" value, selected from a Gaussian distribution centered at zero, with $\sigma = \sigma_t$, is added to the proper time t_p. This smearing is intended to represent the time-resolution limitations of the detector.

2. The new, smeared time, is then smeared again with a second Gaussian centered at zero, but having a lifetime-dependent[1] $\sigma = \sigma_n t_p$. This second smearing is intended to account for unmeasured neutrinos in the decay of the B_s^0, which carry away a proportionally larger amount of momentum for long-lifetime B_s^0 than for short-lifetime B_s^0, regardless of the decay mode[2], thus adding more error to the determination of the decay length of longer-lifetime B_s^0. The final, "measured" time, is then t.

3. Next, it is decided whether or not the B_s^0 in an event "mixes" by comparing a number selected from a flat random distribution between 0 and 2 with the value of $1 + \cos(\Delta m t_p)$ (the mixing comparison), where Δm is the mass difference between the B_s^0 and $\bar{B}_s^0$. If the first number is larger than the mixing comparison, the B_s^0 has mixed to a $\bar{B}_s^0$ otherwise it remains "unmixed". In either case, it is tagged accordingly.

4. Finally, the possibility of mistagging is considered, at a percentage rate[3] α. This is the percentage of events for which a B_s^0 is actually supposed to be a $\bar{B}_s^0$ or vice versa. A number is selected from a flat random distribution between 0 and 1, and compared to α to determine whether the particle is mistagged, smaller numbers than α indicating that this is the case. The tag of the particle is then adjusted accordingly.

To run this simulation, the user must specify:

`tSigma`: σ for time-independent smearing	`nEvts`: The number of events	`dm`: Δm
`nSigma`: σ for time-dependent smearing	`misTag`: The mistag rate (α)	`tau`: τ

The MC is also currently capable of generating a simple exponential background, with a user-definable signal-background ratio, but we have not used this functionality yet.

3 The Fitting

Each set of generated events is histogrammed by lifetime, depending on the tag, and fit with a function[4] which takes the following as parameters: Δm, σ_t, τ, `tag`, α, and t, where t is the independent variable. The actual fit function is given by Equation 2, and uses RooFit [3] for computation of the complex error function.

The code used for fitting (`unbinFitoscd.cpp`) utilizes Root's version of MINUIT [4] to do an unbinned likelihood fit, meaning that generated values of the lifetime t are compared to the distribution function f individually (and therefore independent of any histogram binning), to determine the probability of selecting that value of t from the distribution f. MINUIT adjusts the fit parameter(s) in small steps, and runs the test against f over again until the probability is maximized, and then returns the final value of the fit parameter(s) and the associated error.

[1] We're in the rest frame of the B_s^0.
[2] This will soon be updated to use a distribution which takes decay mode, among other things, into account.
[3] The dilution $D = 1 - 2\alpha$.
[4] `mll_fit_d()` in `unbinFitosc_d.cpp.`, which refers to `lftmosc_plt_d()` and `mixing()` in `func.cpp`

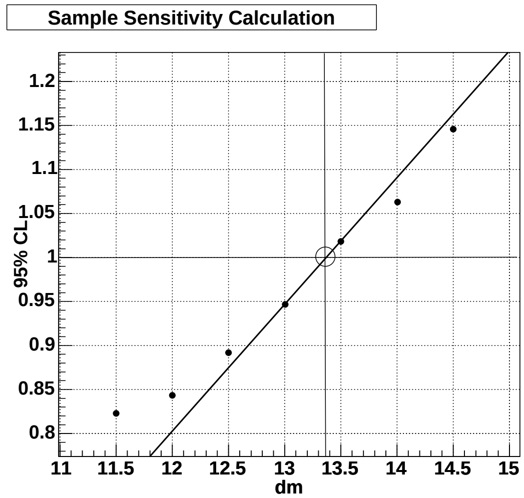

Figure 1: A sample sensitivity calculation. The sensitivity is defined to be the value of Δm for which the error in D is 1. *(Dots: Step 1 and 2; Line: Step 3, Cricle: Step 4.)*

By "fixing" all parameters but α during the fit, we have essentially fit for only the amplitude of the oscillation, hence the "unbinned likelihood amplitude" fitting.

4 Calculating Sensitivity

For each set of conditions tested, we calculated the sensitivity of a measurement of dilution ($D = 1 - 2\alpha$) to the conditions as follows:

1. The process of generating events and fitting for the dilution was done 20+ times for each run, each time with $\Delta m = 1000$ in the MC, and Δm in the fit fixed at a value varied between 0 and ~20 over the course of each run[5].

2. The *error* in α, as reported by MINUIT, was plotted against Δm. *(The dots in Figure 1.)*

3. The equation of a straight line crossing through the dilution error value just below 1, and that just above 1, was determined.*(The line in Figure 1.)*

4. The *Sensitivity* was then calculated to be the value of Δm for which this line crossed 1. *(The open circle in Figure 1.)*

In general, testing the sensitivity of a dilution measurement involved setting one or more parameters differently in the MC than in the fit.

5 Results

Sensitivity of a dilution measurement was tested under the following conditions: Variations in σ_t, variations in D, and variations in σ_n. In the first two cases, the Condor batch scheduler was used to run the series of tests on the OUHEP cluster [5], and in the last case, PBS was used to run the series of tests on OSCER [6]. Each sensitivity value represents a run of 20 generate-and-fit iterations, with Δm varied between 0 and ~20 over the course of the run, as described in Section 4.

[5]Values other than 0-20 were used when needed for reasonable precision.

Table 1: Sensitivity of D to $\pm$ 20% variations in σ_t

Fit ↓ \ MC →	0.08	0.10	0.12
0.08	22.07	23.56	-
0.10	17.66	17.74	17.88
0.12	-	15.74	15.00

Table 2: Sensitivity of D to $\pm$ 20% variations in D

Fit ↓ \ MC →	0.128	0.160	0.192
0.128	16.35	16.35	-
0.160	17.75	17.74	17.75
0.192	-	18.90	18.91

5.1 Variations in σ_t

The effect of $\pm$ 20% variations in σ_t on the sensitivity of D are enumerated in Table 1. Here, the dilution was set to 0.16 both in the MC and as an initial value in the fit, and time-dependent smearing was not used. σ_t was set to values of 0.1 $\pm$ 20% in the MC, and for each of these, fitting was done with σ_t values of 0.1 $\pm$ 20%. The combinations for which the listed sensitivity is "-" were not tested.

5.2 Variations in D

The effect of $\pm$ 20% variations in the value of D in the MC vs. the initial value of D used in fitting, on the sensitivity of a measurement of D, are listed in Table 2. Here, σ_t was set to 0.10 both in the MC and in the fit, and time-dependent smearing was not used. D was set to values of 0.16 $\pm$ 20% in the MC, and for each of these, fitting was done with D at an initial values of 0.16 $\pm$ 20%. The combinations for which the listed sensitivity is "-" were not tested.

5.3 Variations in σ_n

Table 3 lists sensitivity of a measurement of D to $\pm$ 20% variations in σ_n vs. variations in accounting for σ_n in fitting[6]. Although the MC makes it easy to include the time-dependent smearing σ_n, this extra Gaussian

[6] $\sigma_{eff} = \sqrt{\sigma_t^2 + (\sigma_n t)^2} = 0.14$ for $\sigma_t = 0.10$ and $t = 0.6$, as experimentally determined.

Table 3: Sensitivity of D to $\pm$ 20% variations in σ_n

Fit ↓ \ MC →	0.130	0.163	0.196
σ_n ignored	17.799	17.783	17.808
Avg.(σ_n, σ_t)	6.473	5.840	5.162
Eff. σ at t	-	13.386	-

is not so easily included into the fit function. Therefore, we chose three methods of accounting for this extra factor in the fitting. In the first row of Table 3, the time-dependent smearing is ignored altogether and $\sigma_{eff} = \sigma_t$. In the second row, the lifetime values from the MC were histogrammed, and then a bin-weighted average was used to determine an effective (average) value of σ to use in the fit. In the last row, the actual effective smearing at each lifetime t was computed during fitting, using $\sigma_n = 0.163$. In all cases, σ_t was set to 0.1 in the MC and fit, and D was set to 0.16 both in the MC and for the initial fit value.

References

[1] The Root Data Analysis Framework, http://root.cern.ch/.

[2] CERNLib C335: Complex Error Function, http://wwwasdoc.web.cern.ch/wwwasdoc/shortwrupsdir/c335/top.html.

[3] The RooFit Toolkit for Data Modeling *(Contains a direct C++ translation of CWERF for use with Root)*, http://roofit.sourceforge.net/.

[4] CERNLib (PackLib) Long Writeup D506: MINUIT Minimization Package, http://wwwasdoc.web.cern.ch/wwwasdoc/WWW/minuit/minmain/minmain.html.

[5] Information about the OUHEP cluster: http://www-hep.nhn.ou.edu/d0/grid/.

[6] OU Supercomputing Center for Education and Research, http://www.oscer.ou.edu/.