Trilateration

John C. Eidson

Research Memo
March 21, 2012

University of California at Berkeley
Berkeley, CA, 94720, USA

Abstract

This note describes a trilateration application, the applicable equations, and a Ptolemy simulation of the application.

1 Trilateration Basics

Figure 1 illustrates an application for the detection and location of a signal source (T in the figure) based on the differences in the time of receipt of a signal transmitted by the source at three detectors (D1, D2, and D3) whose positions are known. Obviously to do this each detector must contain a local clock accurately synchronized to its peers. The timestamps generated upon receipt are then processed using a trilateration computation (TRI) to obtain the position of the target and the time of transmission. This process is used in gunshot location by local police and for detection of spurious or clandestine rf transmitters, e.g. spy bugs, by civil and military groups. The transmitted signals are sound and electromagnetic, (EM), waves respectively. The detectors typically communicate with each other and the control center where the trilateration computation is performed using a LAN.

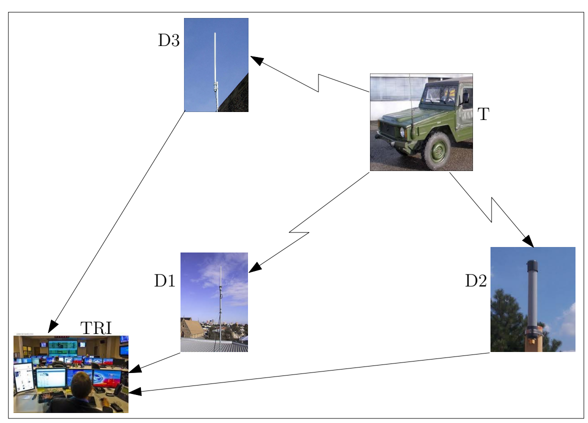

Figure 1. Trilateration Application.

The geometry of this application is illustrated in Figure 2. The three detectors are at fixed locations on the coordinate axes at coordinates $(0, 0, \tau_1)$, $(a, 0, \tau_2)$ and $(b, 0, \tau_3)$ for detectors 1, 2, and 3 respectively where τ_0, τ_1, τ_2, and τ_3 represent respectively the time of signal transmission at target T and the times of reception at detectors D1, D2, and D3. From this geometry the Equations 1, 2, and 3 can be derived where v is the velocity of propagation of the transmitted signal. The unknowns are of course x,y, and τ_0. The solution for these equations is derived in Section 3.

$$r_1^2 = [v(\tau_1 - \tau_0)]^2 = x^2 + y^2 \tag{1}$$

$$r_2^2 = [v(\tau_2 - \tau_0)]^2 = (x-a)^2 + y^2 \tag{2}$$

$$r_3^2 = [v(\tau_3 - \tau_0)]^2 = x^2 + (y-b)^2 \tag{3}$$

In the Ptolemy simulation of this application the velocities are taken to be 300 m/s for sound and $3x10^8$ m/s for electromagnetic radiation. This simulation is discussed further in Section 2.

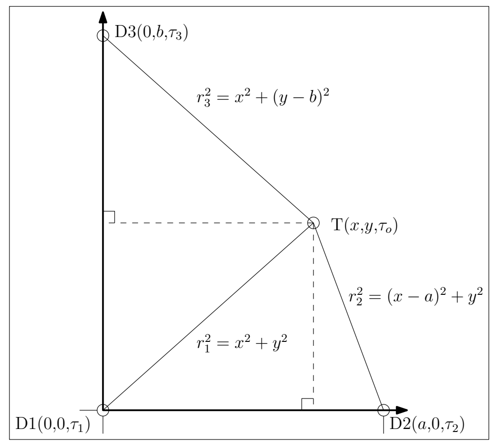

Figure 2. Trilateration Geometry.

In practice things are considerably more complex. Most fielded systems use more than three detectors to provide better coverage. In general this means that rather than a closed for solution to the trilateration equations, some sort of least square fit of multiple measurement data is used. In addition information other than time of receipt, for example signal strength or directional information is often used in the computations. The actual detection of the signals at the detectors is also quite complicated. The received waveforms are far from a clean step function. It is difficult to develop suitable algorithms for determining the time of receipt that is consistent across all detectors and in the face of signal degradation.

It is also critical that the system be robust with respect to timing constraints. For example in the simulation of the geometry of Figure 2 the maximum differential time of signal propagation will occur when the target is close to either detectors 2 or 3. In the simulation $a = 3000$m and $b = 4000$m so with the target say at D2, $r_3 = 5000$m and $\tau_3 = 5000/(3 * 10^8) \approx 16.7\mu$s for an EM signal and 16.7s for sound. If the target signals are transmitted more frequently than every 16.7μs for EM or 16.7s for sound, great care must be take to ensure that the trilateration computation uses receipt times for the same transmission as, in this case for example, D2 will receive a second signal before the previous signal has been received at D3.

If we assume that the desired position accuracy is approximately 10m then intuitively the clocks must be synchronized and timestamps generated with an accuracy of $10/velocity$ or roughly 33ns for EM signals and 33ms for sound signals.

2 Ptolemy Trilateration Simulation

This applications has been simulated using the Ptolemy system. The top level Ptolemy model is illustrated in Figure 3. The blocks within the red rectangle represent the application system. The *Target Signal Generator* block represents the target at the unknown location. This block periodically generates simulated target transmissions at time τ_{0i}, a triplet of events with model times $\tau_0 + r_1/v$, $\tau_0 + r_2/v$, and $\tau_0 + r_3/v$ with the delays r_i/v representing the time of propagation to the three *Target Signal Detectors*. Each event is processed by the appropriate detector block which generates a timestamp upon receipt and forwards the timestamp to the *Trilateration* block. The remaining blocks generate statistics and plots of the simulated performance of the system under different conditions specified by the user via the parameters of the *System Parameters* block.

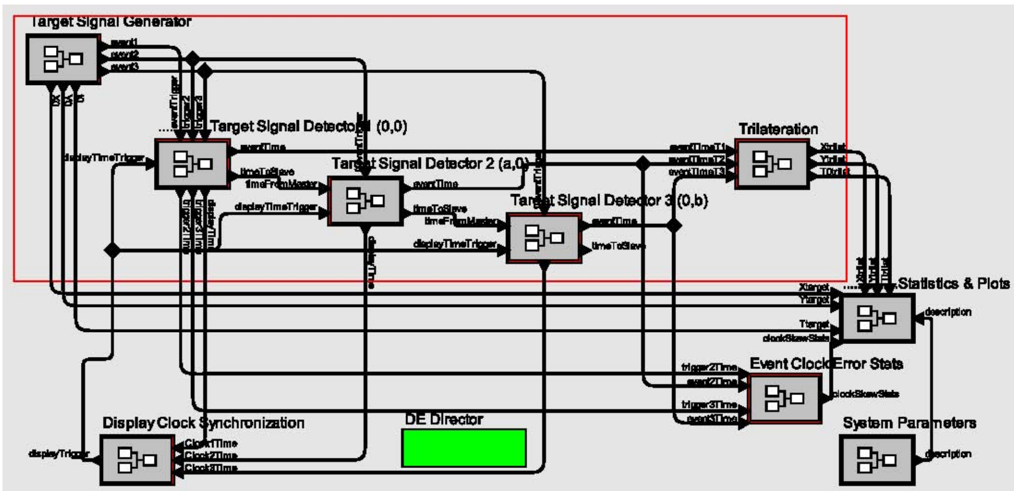

Figure 3. Trilateration Ptolemy Model

The adjustable parameters of the *System Parameters* block are:

- sound: When true, the signal is assumed to be sound and the velocity parameter is set to 300m/s. When false, EM signals are assumed and the velocity is set to $3 * 10^8$m/s.

- MovingTarget: When true, the target transmits signals from *pathPoints* points along the parabola $y = (3.6b/a^2) * (x - a/2)^2$ with points equally distributed on the x axis such that $0 \leq x \leq a$. The path is shown in Figure 4. When false, *pathPoints* transmission all are sent from the point $(a, 0.9b)$. This point is circled on the path in Figure 4. In each case the time between transmission is approximately $sqrt(a^2 + b^2)/velocity + 2$ seconds.

- Synchronized: When true, the clocks in the three detectors are synchronized with Detector 1 serving as the master (and itself free running) and with D2 slaved to D1 and D3 slaved to D2. When false, all three clocks are free running.

- GoodClock: When true (and PerfectClock is false), the values for the free running drift rate is set to 0, 1PPM, and -1PPM for the clocks in detectors 1, 2, and 3 respectively. The 1PPM (0.000001s/s) value is determined by the fixed parameter *goodClockDrift*. The one sigma value for the noise in the time maintained by the clocks is set to 8ns for all clocks. This value is determined by the fixed parameter *goodClockOneSigmaSignalImpairment*. When false (and PerfectClock is false), the corresponding drift rates are 0, 10PPM, and -10PPM and the noise is set to 40ns.

- PerfectClock: When true (irrespective of the value of GoodClock) the drifts and the noise are set to zero for all clocks. When false the drift and noise parameters are set based on the value of the GoodClock parameter.

In addition there is a set of fixed parameters visible from this block as follows:

- goodClockOneSigmaSignalImpairment: default value 8ns. This is the one sigma value of the random distribution of timing impairments implemented in the *Controller* block internal to the *Target Signal Detector* blocks. This controller in turn sets the rates of the actual clock of the DE director to produce statistical variations in the timescale with a nominal normal distribution with this value of standard deviation. See GoodClock parameter above. 8ns is achievable but typically requires more expensive quartz crystals for the clock oscillator and careful attention to clock LSB and servo characteristics. 40ns if readily achievable with care but does not require excessively expensive components.

- goodClockDrift: default value 1PPM. The value for the rate at which a free running clock departs from oracle time. See GoodClock parameter above. 1PPM typically requires better and more expensive oscillators often oven controlled. 10PPM requires less care but still more than the 100PPM of cheap crystal oscillators.

- pathPoints: default value 15. This determines the number of signal transmissions generated.

- a: default value 3000m. Detector 2 is located at (a,0).

- b: default value 4000m. Detector 3 is located at (0,b).

- eventStartTime: default value 600s. This is the time the first signal is transmitted. See discussion of Figure 6 below.

Next examine examples of the various outputs of the simulation. Unless otherwise indicated these examples are taken with the follow parameter values: sound = false, MovingTarget = true, Synchronized = true, GoodClock = true, PerfectClock = false.

Figure 4 shows the computed and actual positions of the target as it moves along the 15 points on the parabola. The actual positions are shown in red and the computed positions in blue. The inset is an enlargement of the area around the third point and reveals a computed position error of about 10m. The locations of the three detectors are

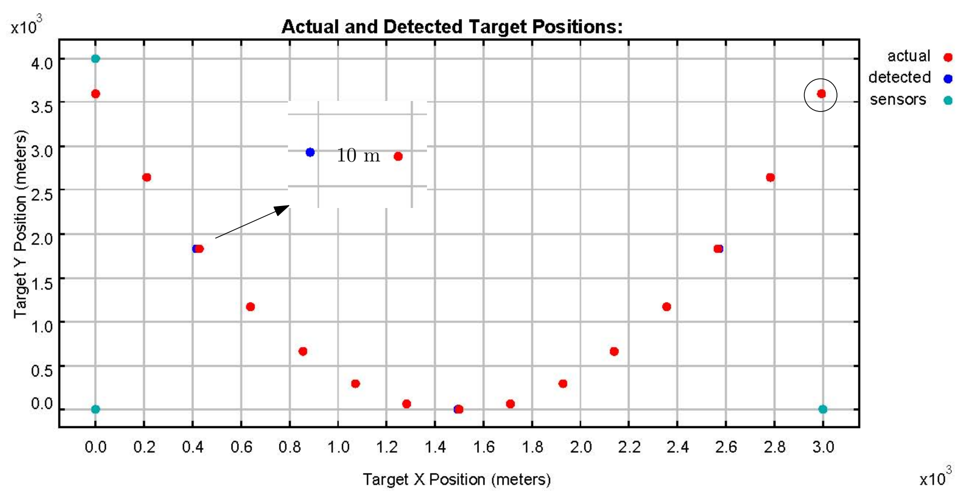

Figure 4. Target Position Plot (moving)

shown as green dots in the figure.

Figure 5 again shows the actual and computed positions but with the parameter MovingTarget set to false. In this case the actual target position for all 15 points is at (a,0.9b), i.e. the circled point in Figure 4. Note that the dimensions on the axes cover approximately 16 and 7 meters along x and y respectively. The computed points are concentrated within this area. The distribution of these points will be discussed later in this section.

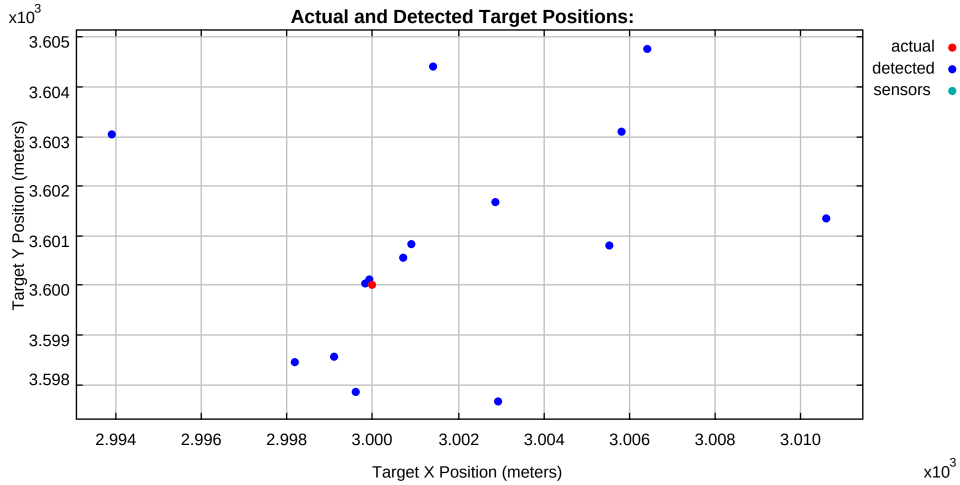

Figure 5. Target Position Plot (fixed)

Figure 6 illustrates the synchronization performance of the clocks in the three detectors during the course of the roughly 950 seconds of the simulation. The $3\mu s$ transient starting at 0 is due to the PID servos in clocks 2 and 3 tracking their parent clocks. While all clocks start at the same time (0) recall that the drift parameters of clocks 2 and 3 are +1PPM and -1PPM respectively. The transient is a result of the integral term in the PID needing to accumulate sufficient rate offset to compensate for the free running drifts of each clock. The transient appears to be settled in about 3 minutes (a typical value for synchronization proto-

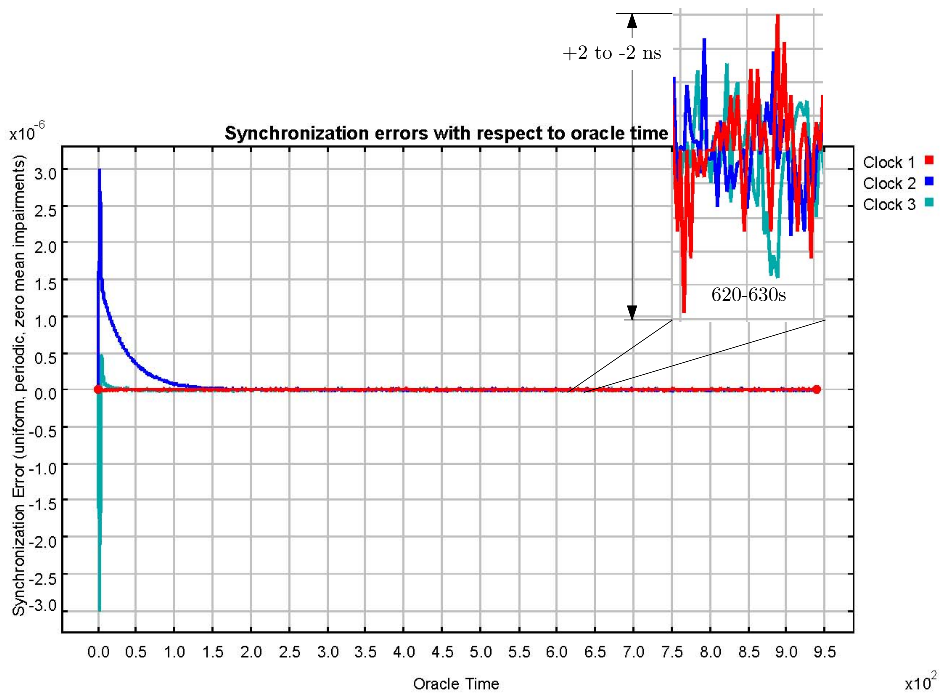

Figure 6. Clock Synchronization

cols) however it takes considerably longer to settle within the errors allowed which depend on the desired positional accuracy and the velocity of propagation. Recall from the discussion at the end of Section 1 that a reasonable value for this error is on the order of 33ns. This transient behavior is why the fixed parameter *eventStartTime* is set to 600s. If detection is started earlier then one or more of the clocks will exhibit offsets due to this transient which will of course degrade the computed position accuracy. The inset in Figure 6 shows the synchronization details for the period from 620 to 630 s. The simulated noise characteristics during this interval span roughly ± 2ns.

Finally Figure 7 shows the display of the most relevant system parameter values and computed statistics. The first paragraph "System Conditions" lists the system parameters selected by the user. The first line reminds us of the locations of the detectors. The second line gives the characteristics of the target– in this case a moving target emitting EM signals. The third line gives the characteristics of the clocks– in this case synchronized clocks with good quality.

System Conditions:
{ = "Sensors at (0,0),(a,0),(0,b) where", a = 3000.0, b = 4000.0}
{ = "Target at X,Y.", Target emission = "electromagnetic", Target is = "moving"}
{ = "Clock", Quality = "good clock", Status = "Synchronized: 1(master,free run)=>2=>3"}

Computed vs Actual Position Error Statistics (meters)
{Mean = 2.870609109026, StdDev = 3.0494159691617}

Computed vs Actual Target Transmit Time Error Statistics (seconds)
{Mean = -1.334835057302068E-9, StdDev = 9.087714367996645E-9}

Clock2 and Clock3 Skew Statistics wrt Clock1 at Event Times (seconds)
{Clock2 Mean = -2.186660215860077E-9, StdDev = 9.538168767693523E-9}
{Clock3 Mean = 9.000132195069455E-10, StdDev = 9.364676247889561E-9}

Figure 7. Simulation Data

The second paragraph "Computed vs Actual Position Error Statistics" gives the mean and standard deviation of the difference in meters between the actual and computed location of the target. In this case the values are each roughly 3 meters indicating that the transient discussed in connection with Figure 6 had settled sufficiently.

The third paragraph "Computed vs Actual Target Transmit Time Error Statistics" gives the mean and standard deviation of the difference in the computed and actual values of the times τ_{0i} when the signals were transmitted. While it is difficult to exactly state the sensitivity of computed position to errors in the computed transmit time note that the standard deviation of the computed times multiplied by the velocity is the same order as the computed deviations in position.

The final paragraph "Clock2 and Clock3 Skew Statistics wrt Clock1 at Event Times" gives the mean and standard deviation of the differences between the time or receipt of a signal on clocks 2 or 3 to the time on clock 1 at this same instant. In other words a measure of the clock synchronization and noise induced errors between the clocks at the times of measurement. Again these times are consistent with the statistics of the computed time differences and position errors noted in Figure 5.

Figure 8 presents simulation data for the case where all clock drifts are zero. Not surprisingly the Clock 2 and 3 skew statistics are both zero in this case since with zero drift the clocks synchronize perfectly and no errors between the clocks is expected. However note that the statistics on the signal transmit time and the position errors are not zero although considerably (five orders of magnitude) less than the corresponding values given in Figure 7. The cause of these residual errors is due to arithmetic precision errors in the trilateration computations. These appear even in the case of a stationary target where the target does not move but the times of transmission of course increase by roughly 2 seconds between each transmission. As will be seen in Section 3, the solution to the equations involves, among other things, differences between transmit times, differences between the squares of these times and can therefore be expected to produce low level errors due to lack of precision, i.e. sufficient significant bits in the representations, in these operations. Finally Figure 9 gives the results for conditions identical to those of Figure 7 except that the signals are assumed to be sound rather than EM. The statistics on the clock performances for the two cases is very similar. Likewise the computed values of signal transmission times have similar statistics for the two cases. The statistics for the position errors for the sound case are roughly six orders of magnitude less than in the EM case perhaps not a surprise since roughly speaking, a given time error is multiplied by the velocity to yield a position error and the velocity of sound and EM waves differ by six orders of magnitude.

System Conditions:
{ = "Sensors at (0,0),(a,0),(0,b) where", a = 3000.0, b = 4000.0}
{ = "Target at X,Y.", Target emission = "electromagnetic", Target is = "stationary"}
{ = "Clock", Quality = "perfect clock", Status = "Synchronized: 1(master,free run)=>2=>3"}

Computed vs Actual Position Error Statistics (meters)
{Mean = 6.938234835434112E-4, StdDev = 1.06935708801947E-5}

Computed vs Actual Target Transmit Time Error Statistics (seconds)
{Mean = -2.334369734550516E-12, StdDev = 5.870763055716148E-14}

Clock2 and Clock3 Skew Statistics wrt Clock1 at Event Times (seconds)
{Clock2 Mean = 0.0, StdDev = 0.0}
{Clock3 Mean = 0.0, StdDev = 0.0}

Figure 8. Simulation Data for Zero Drift

System Conditions:
{ = "Sensors at (0,0),(a,0),(0,b) where", a = 3000.0, b = 4000.0}
{ = "Target at X,Y.", Target emission = "sound", Target is = "moving"}
{ = "Clock", Quality = "good clock", Status = "Synchronized: 1(master,free run)=>2=>3"}

Computed vs Actual Position Error Statistics (meters)
{Mean = 2.075760807623115E-6, StdDev = 1.533790742136426E-6}

Computed vs Actual Target Transmit Time Error Statistics (seconds)
{Mean = 5.994934326736256E-10, StdDev = 5.261402887803933E-9}

Clock2 and Clock3 Skew Statistics wrt Clock1 at Event Times (seconds)
{Clock2 Mean = -3.113329209251485E-9, StdDev = 6.757428396728438E-9}
{Clock3 Mean = 1.579996933287475E-9, StdDev = 8.011092656583759E-9}

Figure 9. Simulation Data for Sound Signals

3 Trilateration Equations

The trilateration equations, Equations 1, 2, and 3 derived from Figure 2 are copied here as Equations 4, 5, and 6. The unknowns are τ_0, x and y. Note that these equations are simpler than the general case since two of the detectors are along the x axis (this can always be arranged by a variable transformation), and that the third detector is on the y axis rather than at a general point in space.

$$r_1^2 = [v(\tau_1 - \tau_0)]^2 = x^2 + y^2 \tag{4}$$

$$r_2^2 = [v(\tau_2 - \tau_0)]^2 = (x-a)^2 + y^2 \tag{5}$$

$$r_3^2 = [v(\tau_3 - \tau_0)]^2 = x^2 + (y-b)^2 \tag{6}$$

The equations can be further simplified by a variable translation such that time is measured from τ_1 by substituting for τ_0 as follows:

$$\tau_0 = t_0 + \tau_1 \tag{7}$$

Then Equations 4, 5 and 6 then become:

$$r_1^2 = [v(-t_0)]^2 = d_0^2 = x^2 + y^2 \tag{8}$$

$$r_2^2 = [v(\tau_2 - \tau_1 - t_0)]^2 = (d_2 - d_0)^2 = (x-a)^2 + y^2 \tag{9}$$

$$r_3^2 = [v(\tau_3 - \tau_1 - t_0)]^2 = (d_3 - d_0)^2 = x^2 + (y-b)^2 \tag{10}$$

where:

$$\begin{aligned} d_0 &= vt_0 \\ d_2 &= v(\tau_2 - \tau_1) \\ d_3 &= v(\tau_3 - \tau_1) \end{aligned} \tag{11}$$

Next compute Equation 8 - Equation 9

$$d_0^2 - (d_2 - d_0)^2 = x^2 + y^2 - [(x-a)^2 + y^2] = 2ax - a^2 \tag{12}$$

or

$$x = [d_0^2 - (d_2 - d_0)^2 + a^2]/2a \tag{13}$$

Next substitute Equation 13 into Equation 8

$$y^2 = d_0^2 - [d_0^2 - (d_2 - d_0)^2 + a^2]^2/4a^2 \tag{14}$$

Next compute Equation 8 - Equation 10

$$d_0^2 - (d_3 - d_0)^2 = x^2 + y^2 - [x^2 + (y-b)^2] = 2by - b^2 \tag{15}$$

or

$$y = [d_0^2 - (d_3 - d_0)^2 + b^2]/2b \tag{16}$$

From Equations 14 and 16:

$$\begin{aligned} y^2 = d_0^2 - [d_0^2 - (d_2 - d_0)^2 + a^2]^2/4a^2 = [d_0^2 - (d_3 - d_0)^2 + b^2]^2/4b^2 \\ d_0^2 - [d_0^2 - (d_2 - d_0)^2 + a^2]^2/4a^2 - [d_0^2 - (d_3 - d_0)^2 + b^2]^2/4b^2 = 0 \end{aligned} \tag{17}$$

or

$$d_0^2 - [-d_2^2 + 2d_2d_0 + a^2]^2/4a^2 - [-d_3^2 + 2d_3d_0 + b^2]^2/4b^2 = 0 \tag{18}$$

or

$$d_0^2[1 - d_2^2/a^2 - d_3^2/b^2] - d_0[d_2(a^2 - d_2^2)/a^2 + d_3(b^2 - d_3^2)/b^2] - [(a^2 - d_2^2)^2/4a^2 + (b^2 - d_3^2)^2/4b^2] = 0 \tag{19}$$

Equation 19 can be solved for d_0 using the quadratic formula. x and y are then obtained by substituting d_0 into Equations 13 and 16 respectively. If desired τ_0 can be computed by substitution of d_0/v (from Equation 11) into Equation 7.

It is Equations 19, 13, 16, 11 and 7 that are implemented in the *Trilateration* block of the simulation model.