> $eq1 := delay = \sqrt{x^2 + \left(y + \frac{h}{\tan(elevation)}\right)^2 + h^2} - \frac{h}{\sin(elevation)} - y \cdot \cos(elevation)$

$$eq1 := delay = \sqrt{x^2 + \left(y + \frac{h}{\tan(elevation)}\right)^2 + h^2} - \frac{h}{\sin(elevation)} - y\cos(elevation) \tag{1}$$

> $eq2 := f\_d = -v\_ty \cdot \cos(elevation) - v\_tz \cdot \sin(elevation) + \frac{v\_rx \cdot x + v\_ry \cdot \left(y + \frac{h}{\tan(elevation)}\right) - v\_rz \cdot h}{\sqrt{x^2 + \left(y + \frac{h}{\tan(elevation)}\right)^2 + h^2}}$

$$eq2 := f\_d = -v\_ty\cos(elevation) - v\_tz\sin(elevation) + \frac{v\_rx\, x + v\_ry\left(y + \frac{h}{\tan(elevation)}\right) - v\_rz\, h}{\sqrt{x^2 + \left(y + \frac{h}{\tan(elevation)}\right)^2 + h^2}} \tag{2}$$

> $sys := \{eq1, eq2\}$

$$sys := \left\{delay = \sqrt{x^2 + \left(y + \frac{h}{\tan(elevation)}\right)^2 + h^2} - \frac{h}{\sin(elevation)} - y\cos(elevation), f\_d = -v\_ty\cos(elevation) - v\_tz\sin(elevation) + \frac{v\_rx\, x + v\_ry\left(y + \frac{h}{\tan(elevation)}\right) - v\_rz\, h}{\sqrt{x^2 + \left(y + \frac{h}{\tan(elevation)}\right)^2 + h^2}}\right\} \tag{3}$$

> $sol := solve(sys, \{x, y\}):$

> $all\_x := allvalues(sol[1]):$

> $sol\_x\_1 := simplify(all\_x[1])$

$$sol\_x\_1 := x = \Big(\big(-\cos(elevation)^2\, v\_ty + (-v\_tz\sin(elevation) - f\_d)\cos(elevation) + v\_ry\big)$$
$$\big(-v\_rx^2\,\big(-4\cos(elevation)^3\, delay\, h\, v\_ty\, v\_tz + \big(2\, h\, delay\,(v\_ty - v\_tz)\,(v\_ty + v\_tz)\sin(elev$$
$$- v\_ty - v\_rz)\, h^2 - 4\, delay\, f\_d\, h\, v\_tz + delay^2\,(v\_ty^2 - v\_tz^2)\big)$$

$$\cos(\mathit{elevation})^2 + ((-2\ (\mathit{v\_rz} + \mathit{v\_tz})\ (\mathit{v\_ry} - \mathit{v\_ty})\ h^2 + 4\ \mathit{delay}\ \mathit{f\_d}\ h\ \mathit{v\_ty}$$

$$+ 2\ \mathit{delay}^2\ \mathit{v\_ty}\ \mathit{v\_tz})\ \sin(\mathit{elevation}) - 2\ h^2\ \mathit{f\_d}\ (\mathit{v\_ry} - \mathit{v\_ty}) - 2\ (-2\ \mathit{v\_ty}\ \mathit{v\_tz}$$

$$+ \mathit{v\_rz}\ (\mathit{v\_ry} - \mathit{v\_ty}))\ \mathit{delay}\ h + 2\ \mathit{f\_d}\ \mathit{v\_ty}\ \mathit{delay}^2)\ \cos(\mathit{elevation})$$

$$+ (2\ \mathit{f\_d}\ (\mathit{v\_rz} + \mathit{v\_tz})\ h^2 + 2\ \mathit{delay}\ (\mathit{f\_d}^2 - \mathit{v\_rx}^2 - \mathit{v\_ry}^2 + \mathit{v\_rz}\ \mathit{v\_tz}$$

$$+ \mathit{v\_tz}^2)\ h + 2\ \mathit{f\_d}\ \mathit{delay}^2\ \mathit{v\_tz})\ \sin(\mathit{elevation}) + (\mathit{f\_d}^2 + \mathit{v\_rz}^2 + 2\ \mathit{v\_rz}\ \mathit{v\_tz}$$

$$+ \mathit{v\_tz}^2)\ h^2 + 2\ \mathit{f\_d}\ \mathit{delay}\ (\mathit{v\_rz} + 2\ \mathit{v\_tz})\ h + \mathit{delay}^2\ (\mathit{f\_d}^2 - \mathit{v\_rx}^2 - \mathit{v\_ry}^2$$

$$+ \mathit{v\_tz}^2))\ \sin(\mathit{elevation})^2)^{1/2} + \mathit{v\_rx}^2\ (h\ (\mathit{v\_ry} - \mathit{v\_ty})\ \cos(\mathit{elevation})^3 + ($$

$$-h\ (\mathit{v\_rz} + \mathit{v\_tz})\ \sin(\mathit{elevation}) - \mathit{f\_d}\ h - \mathit{delay}\ \mathit{v\_tz})\ \cos(\mathit{elevation})^2 - (\mathit{v\_ry}$$

$$- \mathit{v\_ty})\ (\mathit{delay}\ \sin(\mathit{elevation}) + h)\ \cos(\mathit{elevation}) + (\mathit{delay}\ \mathit{f\_d} + \mathit{v\_rz}\ h$$

$$+ h\ \mathit{v\_tz})\ \sin(\mathit{elevation}) + \mathit{f\_d}\ h + \mathit{delay}\ \mathit{v\_tz}))\ \Big/\ (\mathit{v\_rx}\ \sin(\mathit{elevation})\ ((\mathit{v\_ty}^2$$

$$- \mathit{v\_tz}^2)\ \cos(\mathit{elevation})^4 + 2\ \mathit{v\_ty}\ (\mathit{v\_tz}\ \sin(\mathit{elevation}) + \mathit{f\_d})\ \cos(\mathit{elevation})^3$$

$$+ (2\ \mathit{f\_d}\ \mathit{v\_tz}\ \sin(\mathit{elevation}) + \mathit{f\_d}^2 - \mathit{v\_rx}^2 - 2\ \mathit{v\_ry}\ \mathit{v\_ty}$$

$$+ \mathit{v\_tz}^2)\ \cos(\mathit{elevation})^2 - 2\ \mathit{v\_ry}\ (\mathit{v\_tz}\ \sin(\mathit{elevation}) + \mathit{f\_d})\ \cos(\mathit{elevation})$$

$$+ \mathit{v\_rx}^2 + \mathit{v\_ry}^2))$$

```
> with(CodeGeneration) :
```

```
> Python(sol_x_1)
```

```
cg = x == ((-math.cos(elevation) ** 2 * v_ty + (-v_tz * math.sin
(elevation) - f_d) * math.cos(elevation) + v_ry) * math.sqrt(-
v_rx ** 2 * (-4 * math.cos(elevation) ** 3 * delay * h * v_ty *
v_tz + (2 * h * delay * (v_ty - v_tz) * (v_ty + v_tz) * math.sin
(elevation) + (v_tz + v_ry - v_ty + v_rz) * (-v_tz + v_ry - v_ty
- v_rz) * h ** 2 - 4 * delay * f_d * h * v_tz + delay ** 2 *
(v_ty ** 2 - v_tz ** 2)) * math.cos(elevation) ** 2 + ((-2 *
(v_rz + v_tz) * (v_ry - v_ty) * h ** 2 + 4 * delay * f_d * h *
v_ty + 2 * delay ** 2 * v_ty * v_tz) * math.sin(elevation) - 2 *
h ** 2 * f_d * (v_ry - v_ty) - 2 * (-2 * v_ty * v_tz + v_rz *
(v_ry - v_ty)) * delay * h + 2 * f_d * v_ty * delay ** 2) *
math.cos(elevation) + (2 * f_d * (v_rz + v_tz) * h ** 2 + 2 *
delay * (f_d ** 2 - v_rx ** 2 - v_ry ** 2 + v_rz * v_tz + v_tz *
* 2) * h + 2 * f_d * delay ** 2 * v_tz) * math.sin(elevation) +
(f_d ** 2 + v_rz ** 2 + 2 * v_rz * v_tz + v_tz ** 2) * h ** 2 +
2 * f_d * delay * (v_rz + 2 * v_tz) * h + delay ** 2 * (f_d ** 2
- v_rx ** 2 - v_ry ** 2 + v_tz ** 2)) * math.sin(elevation) **
2) + v_rx ** 2 * (h * (v_ry - v_ty) * math.cos(elevation) ** 3 +
(-h * (v_rz + v_tz) * math.sin(elevation) - f_d * h - delay *
v_tz) * math.cos(elevation) ** 2 - (v_ry - v_ty) * (delay *
math.sin(elevation) + h) * math.cos(elevation) + (delay * f_d +
v_rz * h + h * v_tz) * math.sin(elevation) + f_d * h + delay *
v_tz)) / v_rx / math.sin(elevation) / ((v_ty ** 2 - v_tz ** 2) *
math.cos(elevation) ** 4 + 2 * v_ty * (v_tz * math.sin
(elevation) + f_d) * math.cos(elevation) ** 3 + (2 * f_d * v_tz
* math.sin(elevation) + f_d ** 2 - v_rx ** 2 - 2 * v_ry * v_ty +
v_tz ** 2) * math.cos(elevation) ** 2 - 2 * v_ry * (v_tz * math.
sin(elevation) + f_d) * math.cos(elevation) + v_rx ** 2 + v_ry *
* 2)
```

```
> sol_x_2 := simplify(all_x[2])
```

$$
\begin{aligned}
&\mathit{sol\_x\_2} := x = \Big(\big(\cos(\mathit{elevation})^2\, v\_ty + (v\_tz \sin(\mathit{elevation}) + f\_d) \cos(\mathit{elevation}) \\
&\quad - v\_ry \big) \\
&\quad \big(\text{-} v\_rx^2 \big(\text{-}4 \cos(\mathit{elevation})^3\, \mathit{delay}\, h\, v\_ty\, v\_tz + \big(2\, h\, \mathit{delay}\, (v\_ty - v\_tz)\, (v\_ty + v\_tz) \sin(\mathit{elev} \\
&\quad - v\_ty - v\_rz)\, h^2 - 4\, \mathit{delay}\, f\_d\, h\, v\_tz + \mathit{delay}^2\, (v\_ty^2 - v\_tz^2) \big) \\
&\quad \cos(\mathit{elevation})^2 + \big(\big(\text{-}2\, (v\_rz + v\_tz)\, (v\_ry - v\_ty)\, h^2 + 4\, \mathit{delay}\, f\_d\, h\, v\_ty \\
&\quad + 2\, \mathit{delay}^2\, v\_ty\, v\_tz \big) \sin(\mathit{elevation}) - 2\, h^2\, f\_d\, (v\_ry - v\_ty) - 2\, (\text{-}2\, v\_ty\, v\_tz \\
&\quad + v\_rz\, (v\_ry - v\_ty))\, \mathit{delay}\, h + 2\, f\_d\, v\_ty\, \mathit{delay}^2 \big) \cos(\mathit{elevation}) \\
&\quad + \big(2\, f\_d\, (v\_rz + v\_tz)\, h^2 + 2\, \mathit{delay}\, \big(f\_d^2 - v\_rx^2 - v\_ry^2 + v\_rz\, v\_tz \\
&\quad + v\_tz^2 \big)\, h + 2\, f\_d\, \mathit{delay}^2\, v\_tz \big) \sin(\mathit{elevation}) + \big(f\_d^2 + v\_rz^2 + 2\, v\_rz\, v\_tz \\
&\quad + v\_tz^2 \big)\, h^2 + 2\, f\_d\, \mathit{delay}\, (v\_rz + 2\, v\_tz)\, h + \mathit{delay}^2\, \big(f\_d^2 - v\_rx^2 - v\_ry^2 \\
&\quad + v\_tz^2 \big) \big) \sin(\mathit{elevation})^2 \big)^{1/2} + v\_rx^2\, \big(h\, (v\_ry - v\_ty) \cos(\mathit{elevation})^3 + (\\
&\quad \text{-} h\, (v\_rz + v\_tz) \sin(\mathit{elevation}) - f\_d\, h - \mathit{delay}\, v\_tz) \cos(\mathit{elevation})^2 - (v\_ry \\
&\quad - v\_ty)\, (\mathit{delay} \sin(\mathit{elevation}) + h) \cos(\mathit{elevation}) + (\mathit{delay}\, f\_d + v\_rz\, h \\
&\quad + h\, v\_tz) \sin(\mathit{elevation}) + f\_d\, h + \mathit{delay}\, v\_tz \big) \Big) \Big/ \Big(v\_rx \sin(\mathit{elevation})\, \big((v\_ty^2 \\
&\quad - v\_tz^2) \cos(\mathit{elevation})^4 + 2\, v\_ty\, (v\_tz \sin(\mathit{elevation}) + f\_d) \cos(\mathit{elevation})^3 \\
&\quad + \big(2\, f\_d\, v\_tz \sin(\mathit{elevation}) + f\_d^2 - v\_rx^2 - 2\, v\_ry\, v\_ty \\
&\quad + v\_tz^2 \big) \cos(\mathit{elevation})^2 - 2\, v\_ry\, (v\_tz \sin(\mathit{elevation}) + f\_d) \cos(\mathit{elevation}) \\
&\quad + v\_rx^2 + v\_ry^2 \big) \Big)
\end{aligned}
$$

```
> Python(sol_x_2)
cg0 = x == ((math.cos(elevation) ** 2 * v_ty + (v_tz * math.sin
(elevation) + f_d) * math.cos(elevation) - v_ry) * math.sqrt(-
v_rx ** 2 * (-4 * math.cos(elevation) ** 3 * delay * h * v_ty *
v_tz + (2 * h * delay * (v_ty - v_tz) * (v_ty + v_tz) * math.sin
(elevation) + (v_tz + v_ry - v_ty + v_rz) * (-v_tz + v_ry - v_ty
- v_rz) * h ** 2 - 4 * delay * f_d * h * v_tz + delay ** 2 *
(v_ty ** 2 - v_tz ** 2)) * math.cos(elevation) ** 2 + ((-2 *
(v_rz + v_tz) * (v_ry - v_ty) * h ** 2 + 4 * delay * f_d * h *
v_ty + 2 * delay ** 2 * v_ty * v_tz) * math.sin(elevation) - 2 *
h ** 2 * f_d * (v_ry - v_ty) - 2 * (-2 * v_ty * v_tz + v_rz *
(v_ry - v_ty)) * delay * h + 2 * f_d * v_ty * delay ** 2) *
math.cos(elevation) + (2 * f_d * (v_rz + v_tz) * h ** 2 + 2 *
delay * (f_d ** 2 - v_rx ** 2 - v_ry ** 2 + v_rz * v_tz + v_tz *
* 2) * h + 2 * f_d * delay ** 2 * v_tz) * math.sin(elevation) +
(f_d ** 2 + v_rz ** 2 + 2 * v_rz * v_tz + v_tz ** 2) * h ** 2 +
2 * f_d * delay * (v_rz + 2 * v_tz) * h + delay ** 2 * (f_d ** 2
- v_rx ** 2 - v_ry ** 2 + v_tz ** 2)) * math.sin(elevation) **
2) + v_rx ** 2 * (h * (v_ry - v_ty) * math.cos(elevation) ** 3 +
(-h * (v_rz + v_tz) * math.sin(elevation) - f_d * h - delay *
v_tz) * math.cos(elevation) ** 2 - (v_ry - v_ty) * (delay *
math.sin(elevation) + h) * math.cos(elevation) + (delay * f_d +
v_rz * h + h * v_tz) * math.sin(elevation) + f_d * h + delay *
v_tz)) / v_rx / math.sin(elevation) / ((v_ty ** 2 - v_tz ** 2) *
math.cos(elevation) ** 4 + 2 * v_ty * (v_tz * math.sin
(elevation) + f_d) * math.cos(elevation) ** 3 + (2 * f_d * v_tz
* math.sin(elevation) + f_d ** 2 - v_rx ** 2 - 2 * v_ry * v_ty +
v_tz ** 2) * math.cos(elevation) ** 2 - 2 * v_ry * (v_tz * math.
sin(elevation) + f_d) * math.cos(elevation) + v_rx ** 2 + v_ry *
* 2)
```

> $all\_y := allvalues(sol[2]) :$

> $sol\_y\_1 := simplify(all\_y[1])$

$$
\begin{aligned}
sol\_y\_1 := y = \Big(& 2 \cos(elevation)^4\, delay\, v\_ty\, v\_tz + \big((-delay\, v\_ty^2 \\
&+ delay\, v\_tz^2) \sin(elevation) + h\, v\_tz^2 + (2\, delay\, f\_d + v\_rz\, h)\, v\_tz \\
&+ h\, v\_ty\, (v\_ry - v\_ty) \big) \cos(elevation)^3 + \big(((v\_ry - 2\, v\_ty)\, h\, v\_tz \\
&- v\_ty\, (2\, delay\, f\_d + v\_rz\, h)) \sin(elevation) - delay\, (v\_ry + 2\, v\_ty)\, v\_tz \\
&+ h\, f\_d\, (v\_ry - 2\, v\_ty) \big) \cos(elevation)^2 + \big((-delay\, v\_tz^2 - 2\, f\_d\, h\, v\_tz \\
&- f\_d\, h\, v\_rz - delay\, (f\_d^2 - v\_rx^2 - v\_ry\, v\_ty)) \sin(elevation) - h\, v\_tz^2 + (\\
&-2\, delay\, f\_d - v\_rz\, h)\, v\_tz - h\, (f\_d^2 + v\_ry^2 - v\_ry\, v\_ty) \big) \cos(elevation) \\
&+ v\_ry\, (delay\, f\_d + v\_rz\, h + h\, v\_tz) \sin(elevation) + f\_d\, h\, v\_ry \\
&+ delay\, v\_ry\, v\_tz \\
&- \big(-v\_rx^2\, \big(-4 \cos(elevation)^3\, delay\, h\, v\_ty\, v\_tz + \big(2\, h\, delay\, (v\_ty - v\_tz)\, (v\_ty + v\_tz) \sin(el \\
&+ 2\, delay^2\, v\_ty\, v\_tz \big) \sin(elevation) - 2\, h^2\, f\_d\, (v\_ry - v\_ty) - 2\, (-2\, v\_ty\, v\_tz \\
&+ v\_rz\, (v\_ry - v\_ty))\, delay\, h + 2\, f\_d\, v\_ty\, delay^2 \big) \cos(elevation) \\
&+ \big(2\, f\_d\, (v\_rz + v\_tz)\, h^2 + 2\, delay\, \big(f\_d^2 - v\_rx^2 - v\_ry^2 + v\_rz\, v\_tz \\
&+ v\_tz^2 \big)\, h + 2\, f\_d\, delay^2\, v\_tz \big) \sin(elevation) + \big(f\_d^2 + v\_rz^2 + 2\, v\_rz\, v\_tz \\
&+ v\_tz^2 \big)\, h^2 + 2\, f\_d\, delay\, (v\_rz + 2\, v\_tz)\, h + delay^2\, \big(f\_d^2 - v\_rx^2 - v\_ry^2 \\
&+ v\_tz^2 \big) \big) \sin(elevation)^2 \big)^{1/2} \Big) \Big/ \big(\sin(elevation)\, \big((v\_ty^2 \\
&- v\_tz^2) \cos(elevation)^4 + 2\, v\_ty\, (v\_tz \sin(elevation) + f\_d) \cos(elevation)^3 \\
&+ \big(2\, f\_d\, v\_tz \sin(elevation) + f\_d^2 - v\_rx^2 - 2\, v\_ry\, v\_ty \\
&+ v\_tz^2 \big) \cos(elevation)^2 - 2\, v\_ry\, (v\_tz \sin(elevation) + f\_d) \cos(elevation) \\
&+ v\_rx^2 + v\_ry^2 \big) \big)
\end{aligned}
$$

> *Python*(*sol_y_1*)

```
cg1 = y == (2 * math.cos(elevation) ** 4 * delay * v_ty * v_tz +
((-delay * v_ty ** 2 + delay * v_tz ** 2) * math.sin(elevation)
+ h * v_tz ** 2 + (2 * delay * f_d + v_rz * h) * v_tz + h * v_ty
* (v_ry - v_ty)) * math.cos(elevation) ** 3 + (((v_ry - 2 *
v_ty) * h * v_tz - v_ty * (2 * delay * f_d + v_rz * h)) * math.
sin(elevation) - delay * (v_ry + 2 * v_ty) * v_tz + h * f_d *
(v_ry - 2 * v_ty)) * math.cos(elevation) ** 2 + ((-delay * v_tz
** 2 - 2 * f_d * h * v_tz - f_d * h * v_rz - delay * (f_d ** 2 -
v_rx ** 2 - v_ry * v_ty)) * math.sin(elevation) - h * v_tz ** 2
+ (-2 * delay * f_d - v_rz * h) * v_tz - h * (f_d ** 2 + v_ry **
2 - v_ry * v_ty)) * math.cos(elevation) + v_ry * (delay * f_d +
v_rz * h + h * v_tz) * math.sin(elevation) + f_d * h * v_ry +
delay * v_ry * v_tz - math.sqrt(-v_rx ** 2 * (-4 * math.cos
(elevation) ** 3 * delay * h * v_ty * v_tz + (2 * h * delay *
(v_ty - v_tz) * (v_ty + v_tz) * math.sin(elevation) + (v_tz +
v_ry - v_ty + v_rz) * (-v_tz + v_ry - v_ty - v_rz) * h ** 2 - 4
* delay * f_d * h * v_tz + delay ** 2 * (v_ty ** 2 - v_tz ** 2))
* math.cos(elevation) ** 2 + ((-2 * (v_rz + v_tz) * (v_ry -
v_ty) * h ** 2 + 4 * delay * f_d * h * v_ty + 2 * delay ** 2 *
v_ty * v_tz) * math.sin(elevation) - 2 * h ** 2 * f_d * (v_ry -
v_ty) - 2 * (-2 * v_ty * v_tz + v_rz * (v_ry - v_ty)) * delay *
h + 2 * f_d * v_ty * delay ** 2) * math.cos(elevation) + (2 *
f_d * (v_rz + v_tz) * h ** 2 + 2 * delay * (f_d ** 2 - v_rx ** 2
- v_ry ** 2 + v_rz * v_tz + v_tz ** 2) * h + 2 * f_d * delay **
2 * v_tz) * math.sin(elevation) + (f_d ** 2 + v_rz ** 2 + 2 *
v_rz * v_tz + v_tz ** 2) * h ** 2 + 2 * f_d * delay * (v_rz + 2
* v_tz) * h + delay ** 2 * (f_d ** 2 - v_rx ** 2 - v_ry ** 2 +
v_tz ** 2)) * math.sin(elevation) ** 2)) / math.sin(elevation) /
((v_ty ** 2 - v_tz ** 2) * math.cos(elevation) ** 4 + 2 * v_ty *
(v_tz * math.sin(elevation) + f_d) * math.cos(elevation) ** 3 +
(2 * f_d * v_tz * math.sin(elevation) + f_d ** 2 - v_rx ** 2 - 2
* v_ry * v_ty + v_tz ** 2) * math.cos(elevation) ** 2 - 2 * v_ry
* (v_tz * math.sin(elevation) + f_d) * math.cos(elevation) +
v_rx ** 2 + v_ry ** 2)
```

> $sol\_y\_2 := simplify(all\_y[2])$

$$sol\_y\_2 := y = \Big(2\cos(elevation)^4\, delay\, v\_ty\, v\_tz + \big(\big(-delay\, v\_ty^2$$

$$+ delay\, v\_tz^2\big)\sin(elevation) + h\, v\_tz^2 + (2\, delay\, f\_d + v\_rz\, h)\, v\_tz$$

$$+ h\, v\_ty\,(v\_ry - v\_ty)\big)\cos(elevation)^3 + \big(\big((v\_ry - 2\, v\_ty)\, h\, v\_tz$$

$$- v\_ty\,(2\, delay\, f\_d + v\_rz\, h)\big)\sin(elevation) - delay\,(v\_ry + 2\, v\_ty)\, v\_tz$$

$$+ h\, f\_d\,(v\_ry - 2\, v\_ty)\big)\cos(elevation)^2 + \big(\big(-delay\, v\_tz^2 - 2\, f\_d\, h\, v\_tz$$

$$- f\_d\, h\, v\_rz - delay\,\big(f\_d^2 - v\_rx^2 - v\_ry\, v\_ty\big)\big)\sin(elevation) - h\, v\_tz^2 + ($$

$$-2\, delay\, f\_d - v\_rz\, h)\, v\_tz - h\,\big(f\_d^2 + v\_ry^2 - v\_ry\, v\_ty\big)\big)\cos(elevation)$$

$$+ v\_ry\,(delay\, f\_d + v\_rz\, h + h\, v\_tz)\sin(elevation) + f\_d\, h\, v\_ry$$

$$+ delay\, v\_ry\, v\_tz$$

$$+ \big(-v\_rx^2\,\big(-4\cos(elevation)^3\, delay\, h\, v\_ty\, v\_tz + \big(2\, h\, delay\,(v\_ty - v\_tz)\,(v\_ty + v\_tz)\sin(el$$

$$+ 2\, \mathit{delay}^2\, v\_ty\, v\_tz)\, \sin(\mathit{elevation}) - 2\, h^2\, f\_d\, (v\_ry - v\_ty) - 2\, (-2\, v\_ty\, v\_tz$$

$$+ v\_rz\, (v\_ry - v\_ty))\, \mathit{delay}\, h + 2\, f\_d\, v\_ty\, \mathit{delay}^2)\, \cos(\mathit{elevation})$$

$$+ (2\, f\_d\, (v\_rz + v\_tz)\, h^2 + 2\, \mathit{delay}\, (f\_d^2 - v\_rx^2 - v\_ry^2 + v\_rz\, v\_tz$$

$$+ v\_tz^2)\, h + 2\, f\_d\, \mathit{delay}^2\, v\_tz)\, \sin(\mathit{elevation}) + (f\_d^2 + v\_rz^2 + 2\, v\_rz\, v\_tz$$

$$+ v\_tz^2)\, h^2 + 2\, f\_d\, \mathit{delay}\, (v\_rz + 2\, v\_tz)\, h + \mathit{delay}^2\, (f\_d^2 - v\_rx^2 - v\_ry^2$$

$$+ v\_tz^2))\, \sin(\mathit{elevation})^2)^{1/2}\Big) \Big/ (\sin(\mathit{elevation})\, ((v\_ty^2$$

$$- v\_tz^2)\, \cos(\mathit{elevation})^4 + 2\, v\_ty\, (v\_tz \sin(\mathit{elevation}) + f\_d)\, \cos(\mathit{elevation})^3$$

$$+ (2\, f\_d\, v\_tz \sin(\mathit{elevation}) + f\_d^2 - v\_rx^2 - 2\, v\_ry\, v\_ty$$

$$+ v\_tz^2)\, \cos(\mathit{elevation})^2 - 2\, v\_ry\, (v\_tz \sin(\mathit{elevation}) + f\_d)\, \cos(\mathit{elevation})$$

$$+ v\_rx^2 + v\_ry^2))$$

```
> Python(sol_y_2)
cg2 = y == (2 * math.cos(elevation) ** 4 * delay * v_ty * v_tz +
((-delay * v_ty ** 2 + delay * v_tz ** 2) * math.sin(elevation)
+ h * v_tz ** 2 + (2 * delay * f_d + v_rz * h) * v_tz + h * v_ty
* (v_ry - v_ty)) * math.cos(elevation) ** 3 + (((v_ry - 2 *
v_ty) * h * v_tz - v_ty * (2 * delay * f_d + v_rz * h)) * math.
sin(elevation) - delay * (v_ry + 2 * v_ty) * v_tz + h * f_d *
(v_ry - 2 * v_ty)) * math.cos(elevation) ** 2 + ((-delay * v_tz
** 2 - 2 * f_d * h * v_tz - f_d * h * v_rz - delay * (f_d ** 2 -
v_rx ** 2 - v_ry * v_ty)) * math.sin(elevation) - h * v_tz ** 2
+ (-2 * delay * f_d - v_rz * h) * v_tz - h * (f_d ** 2 + v_ry **
2 - v_ry * v_ty)) * math.cos(elevation) + v_ry * (delay * f_d +
v_rz * h + h * v_tz) * math.sin(elevation) + f_d * h * v_ry +
delay * v_ry * v_tz + math.sqrt(-v_rx ** 2 * (-4 * math.cos
(elevation) ** 3 * delay * h * v_ty * v_tz + (2 * h * delay *
(v_ty - v_tz) * (v_ty + v_tz) * math.sin(elevation) + (v_tz +
v_ry - v_ty + v_rz) * (-v_tz + v_ry - v_ty - v_rz) * h ** 2 - 4
* delay * f_d * h * v_tz + delay ** 2 * (v_ty ** 2 - v_tz ** 2))
* math.cos(elevation) ** 2 + ((-2 * (v_rz + v_tz) * (v_ry -
v_ty) * h ** 2 + 4 * delay * f_d * h * v_ty + 2 * delay ** 2 *
v_ty * v_tz) * math.sin(elevation) - 2 * h ** 2 * f_d * (v_ry -
v_ty) - 2 * (-2 * v_ty * v_tz + v_rz * (v_ry - v_ty)) * delay *
h + 2 * f_d * v_ty * delay ** 2) * math.cos(elevation) + (2 *
f_d * (v_rz + v_tz) * h ** 2 + 2 * delay * (f_d ** 2 - v_rx ** 2
- v_ry ** 2 + v_rz * v_tz + v_tz ** 2) * h + 2 * f_d * delay **
2 * v_tz) * math.sin(elevation) + (f_d ** 2 + v_rz ** 2 + 2 *
v_rz * v_tz + v_tz ** 2) * h ** 2 + 2 * f_d * delay * (v_rz + 2
* v_tz) * h + delay ** 2 * (f_d ** 2 - v_rx ** 2 - v_ry ** 2 +
v_tz ** 2)) * math.sin(elevation) ** 2)) / math.sin(elevation) /
((v_ty ** 2 - v_tz ** 2) * math.cos(elevation) ** 4 + 2 * v_ty *
(v_tz * math.sin(elevation) + f_d) * math.cos(elevation) ** 3 +
(2 * f_d * v_tz * math.sin(elevation) + f_d ** 2 - v_rx ** 2 - 2
* v_ry * v_ty + v_tz ** 2) * math.cos(elevation) ** 2 - 2 * v_ry
* (v_tz * math.sin(elevation) + f_d) * math.cos(elevation) +
v_rx ** 2 + v_ry ** 2)
>
```