Insertion and matching

Insertion

The driver must enter the dipole that bends the witness into the plasma stages. The first stage is different and does not need a bending magnet, but needs to match both driver and witness.

$$beams \rightarrow matching/focusing \rightarrow plasma\ stage \rightarrow Dipole$$
$$\rightarrow plasmalens \rightarrow chicane \rightarrow plasmalens \rightarrow dipole$$

The old driver is extracted at the first dipole and the new driver is inserted at the last dipole.

Kicks

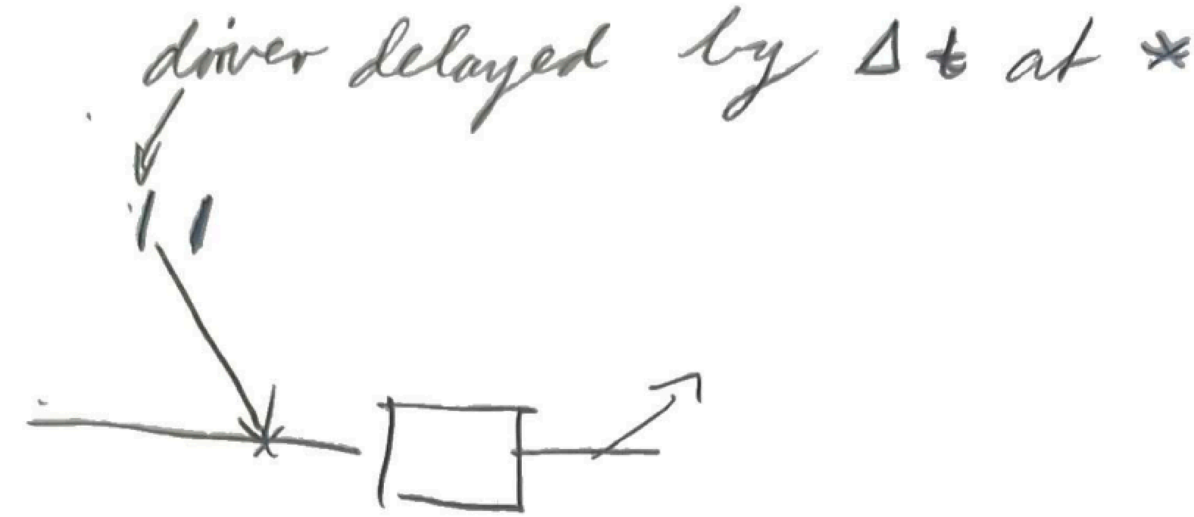
One of the main challenges will be the kicks of the drivers. With N drivers followed by a witness bunch, N-1 drivers must be kicked away without kicking the last beam. Alternatively, if a fast rise kicker is the optimal solution, then we can use this to kick only the first driver and witness into the main beamline, while keeping the rest of the drivers on the trajectory onto the delay-line.

Delay mechanism

For N number of drivers, there are N-1 drivers that needs to be delayed to reach the right phase. The second driver will need to be delayed by the distance between it and the first driver (assuming the first one is allready in the right phase).

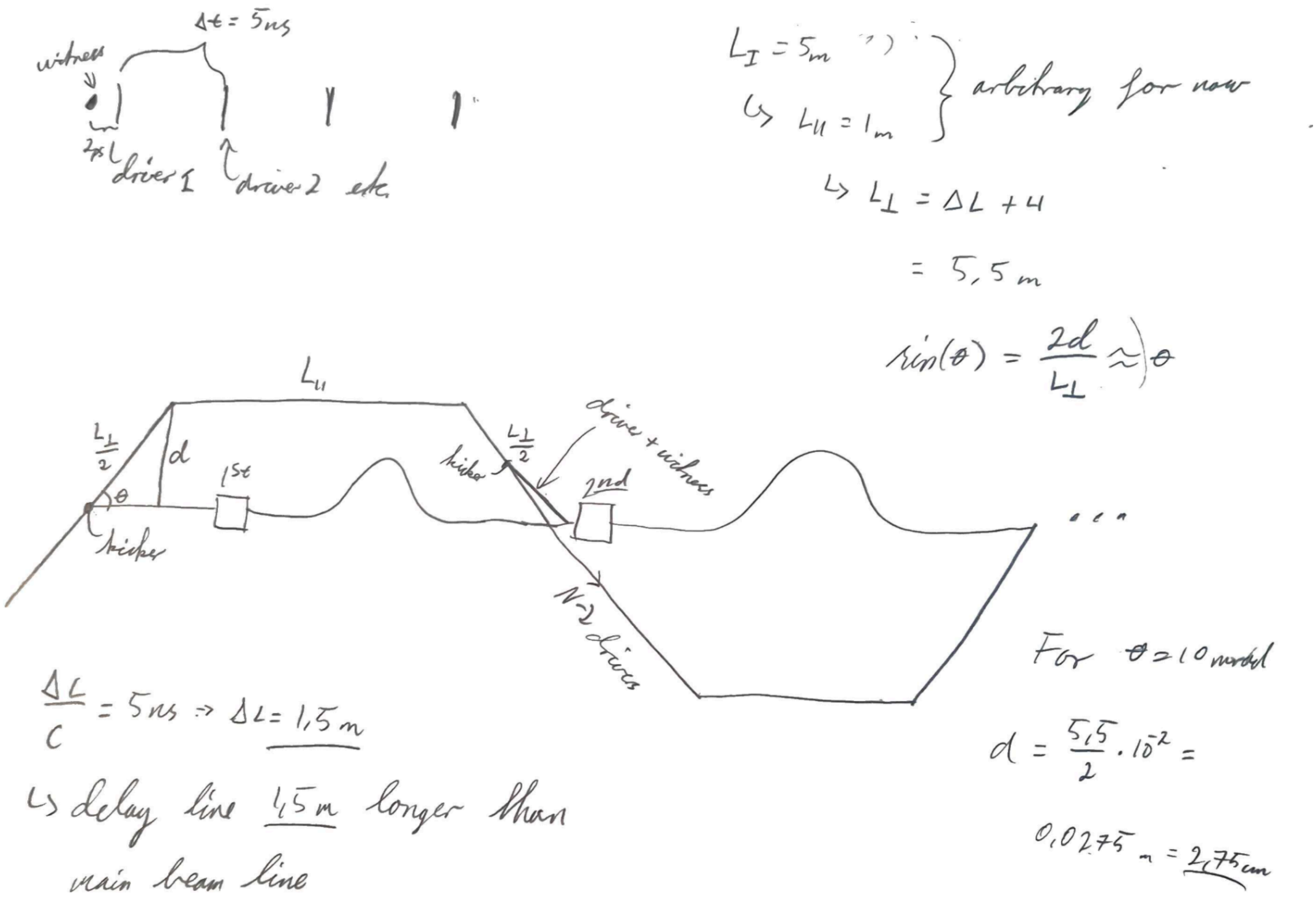
2 drivers

After the first stage there will be 2 drivers left. If we bend just the to-be-used driver will correspond to a proper delay of that driver. But when we do the same in the next stage, that driver will be delayed by the same amount as the first, but it really needs a double delay, since it is twice as long away from the witness.



Multiple drivers

A possibility might be to bend all drivers towards the main beamline, but let the unused drivers remain unused by going straight through. I don't know if this is doable due to wakefields, but since the unused drivers are further away from the witness, there should be no overlap of beams.



$$\Delta L = L_{\parallel} + L_{\perp} - L_I$$

This sketch shows the thought out solution, but is not realistic. We have to take into account that

$$2\frac{L_\perp}{2}cos(\theta) + L_\parallel < L_I,$$

since the prjoection of the main beamline onto the z-axis is less than its length (with the chicanes). For this inequality to hold for the given parameters, θ must be bigger than 0.76 which would make d at least 1.9m.

For a bunch seperation of 5 ns, and a main beamline of 5 m, with no parallel part to the driver delay chicane, then we get the minimum kick required and the resulting $L_\perp$ and d.

```
L_mainline = 5 # m
L_parallel = 0 # m
bunch_separation = 5e-9 # s
delta_L = SI.c*bunch_separation

L_perp = delta_L + L_mainline - L_parallel

theta = np.arccos((L_mainline - L_parallel)/L_perp) # minimum limit

d = L_perp/2 * np.sin(theta)

display(Latex(fr'$\theta = {theta}$, $d = {d}$, $L_\perp = {L_perp}$'))
```

✓ 0.0s

$\theta = 0.69296766140469$, $d = 2.0758438553298055$, $L_\perp = 6.49896229$

This value for d is likely too large for the IR12 tunnel.

For a longer beamline:

```
L_mainline = 9 # m
L_parallel = 0 # m
bunch_separation = 5e-9 # s
delta_L = SI.c*bunch_separation

L_perp = delta_L + L_mainline - L_parallel

theta = np.arccos((L_mainline - L_parallel)/L_perp) # minimum limit

d = L_perp/2 * np.sin(theta)

display(Latex(fr'$\theta = {theta}$, $d = {d}$, $L_\perp = {L_perp}$'))
```

✓ 0.0s

$\theta = 0.5409350249165458$, $d = 2.703155987306413$, $L_\perp = 10.49896229$

For a shorter beamline:

```python
L_mainline = 2 # m
L_parallel = 0 # m
bunch_separation = 5e-9 # s
delta_L = SI.c*bunch_separation

L_perp = delta_L + L_mainline - L_parallel

theta = np.arccos((L_mainline - L_parallel)/L_perp) # minimum limit

d = L_perp/2 * np.sin(theta)

display(Latex(fr'$\theta = {theta}$, $d = {d}$, $L_\perp = {L_perp}$'))
```

✓ 0.0s

$\theta = 0.9623442236638144$, $d = 1.435508368735798$, $L_\perp = 3.49896229$

Conclusion

It seems that we need a smaller beamline and a stronger kick to fit in the IR12 tunnel for this specific driver distribution system.