TIME DELAY INTERFEROMETRY RANGING

Natalia Korsakova
SYRTE/Observatoire Paris

Martin Staab
AEI Hannover/MPI for Gravitational Physics

Aurélien Hees
SYRTE/Observatoire Paris

FIRST TF

SYRTE l'Observatoire de Paris | PSL

SYstèmes de Référence Temps-Espace

TDI ranging

- LISA
- Intro to Time Delay Interferometry
- TDI Ranging
- Sampling method to estimate ranges
- Reinforcement learning method
- Summary

Laser Interferometer Space Antenna

- Gravitational wave detector in space
- Measurement principle in based on laser interferometry
- Three spacecraft on the Earth trailing orbit separated by 2.5 million km
- Sensitive in low frequency compared to ground based detectors
- Planned launch 2034

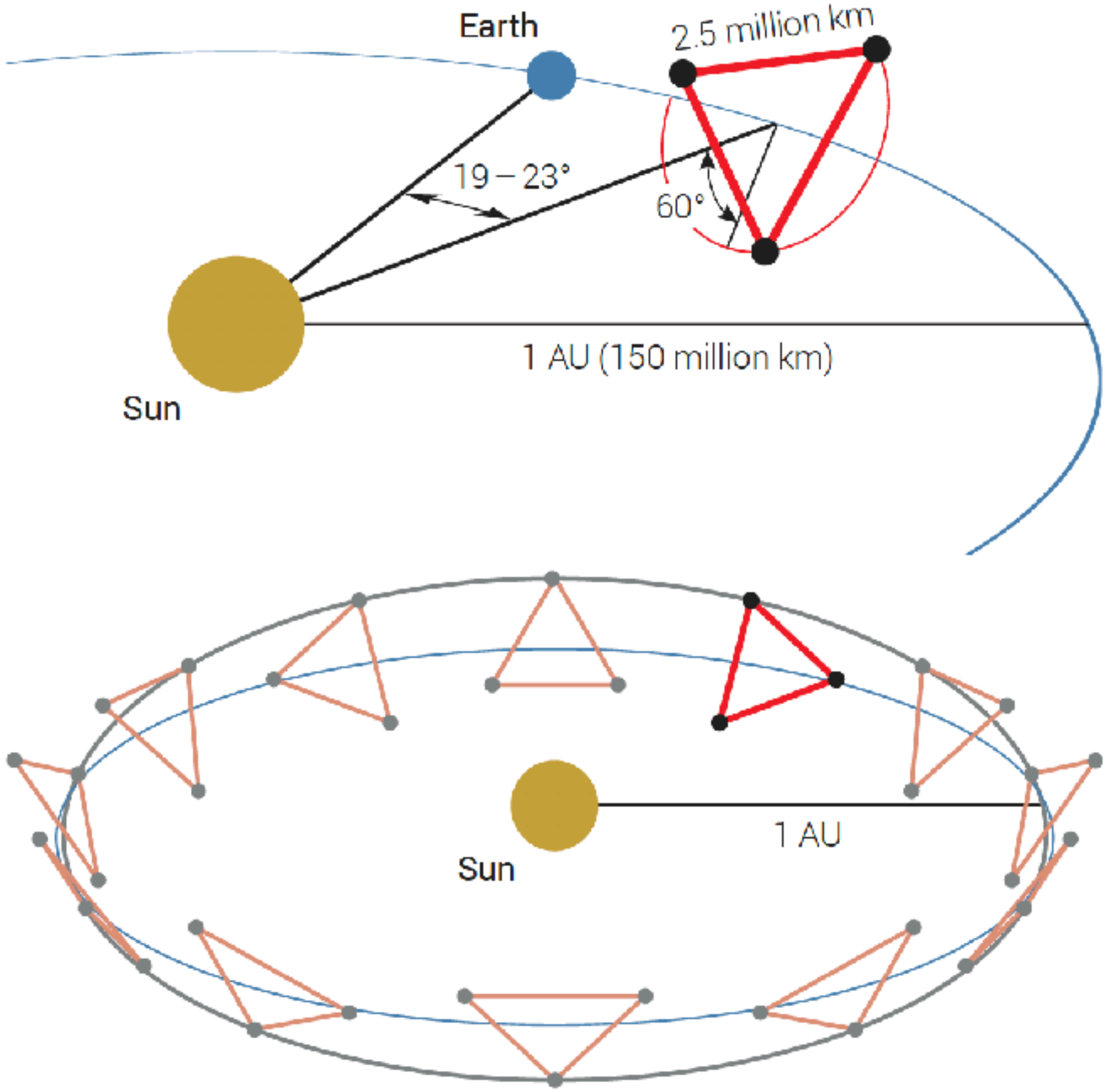


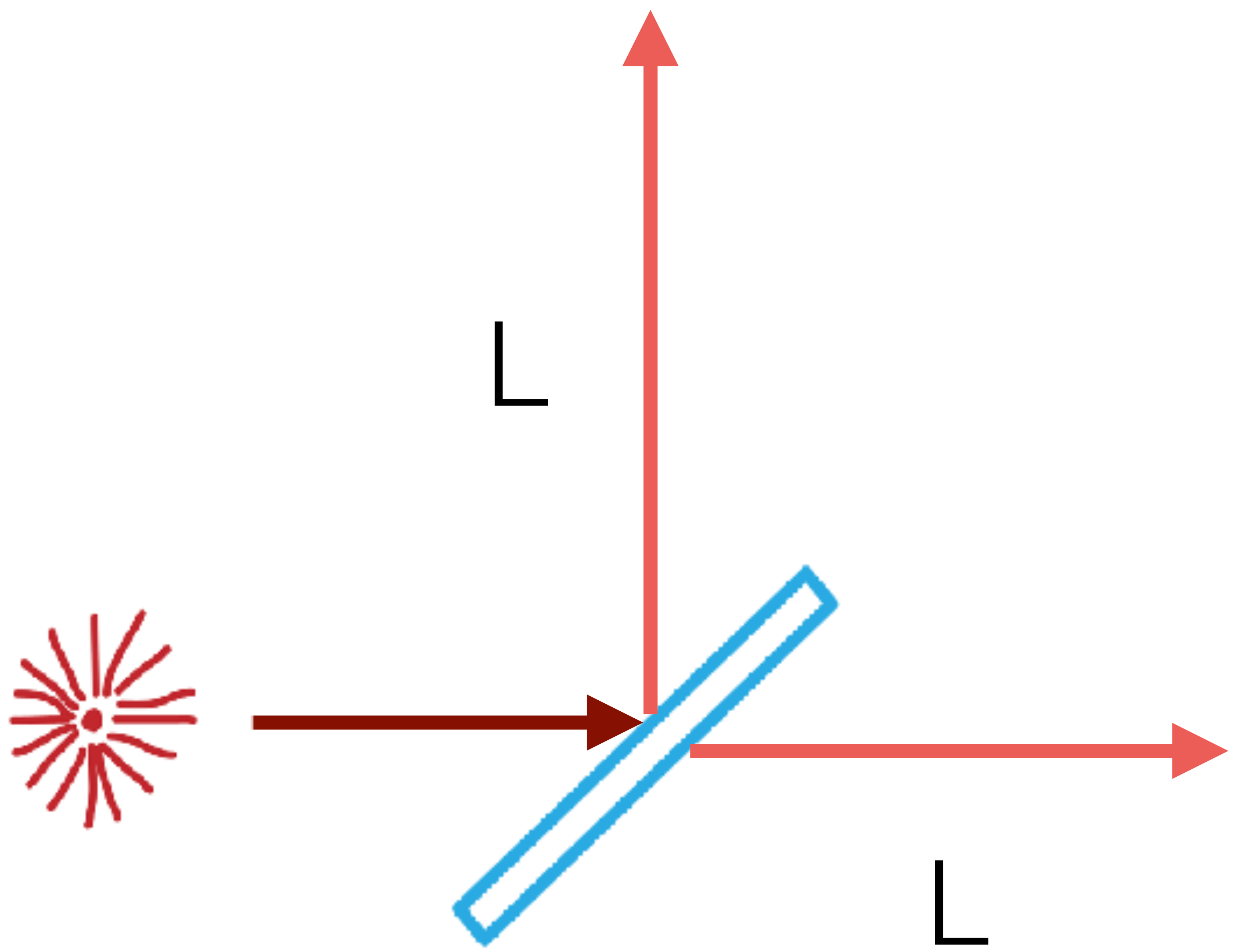
Image: LISA White paper

Laser Interferometry

- Equal-arm interferometer detector

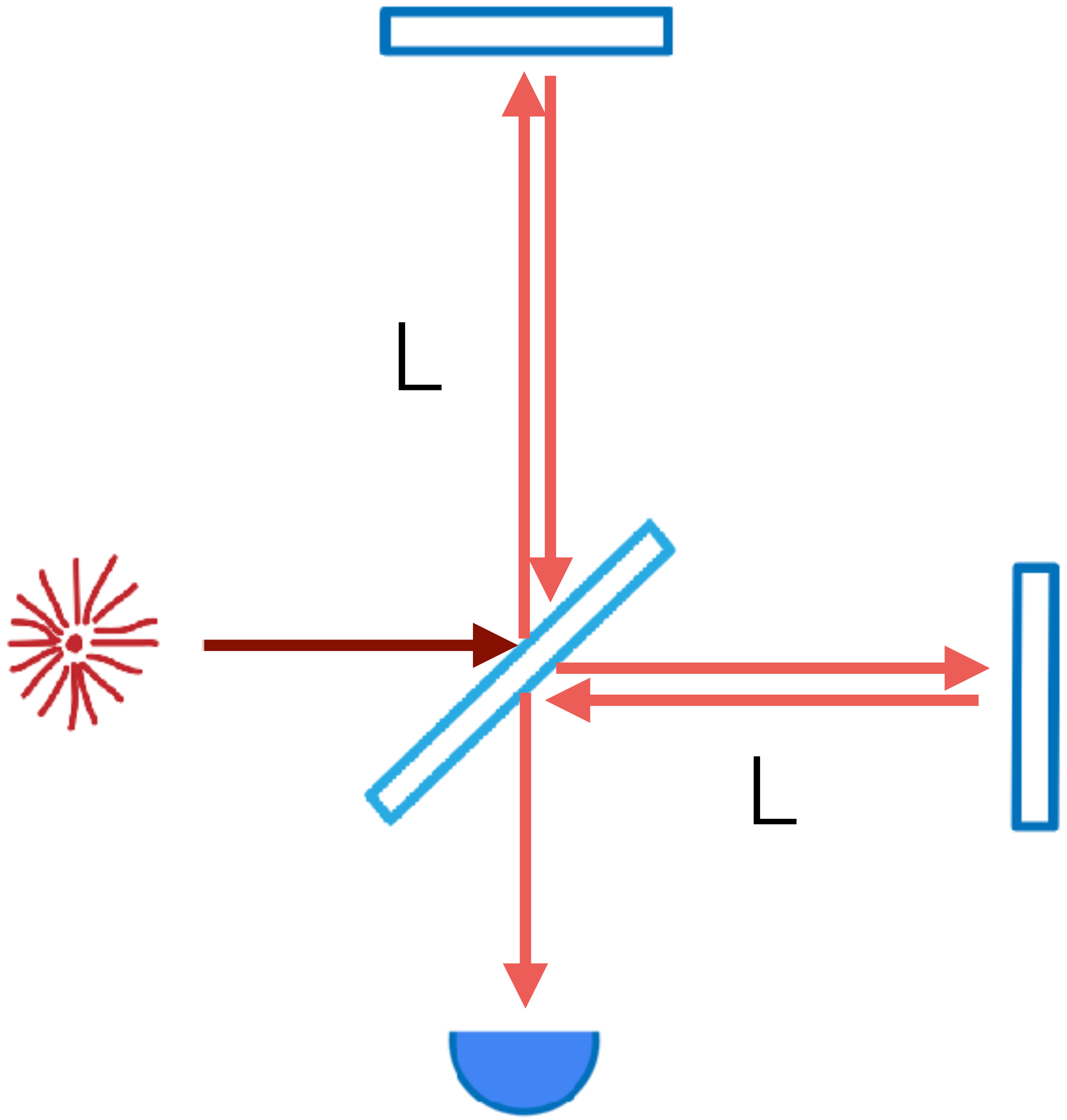
- Cancel laser frequency noise by comparing phases of split beams propagated along the equal arms of the detector.

Laser Interferometry



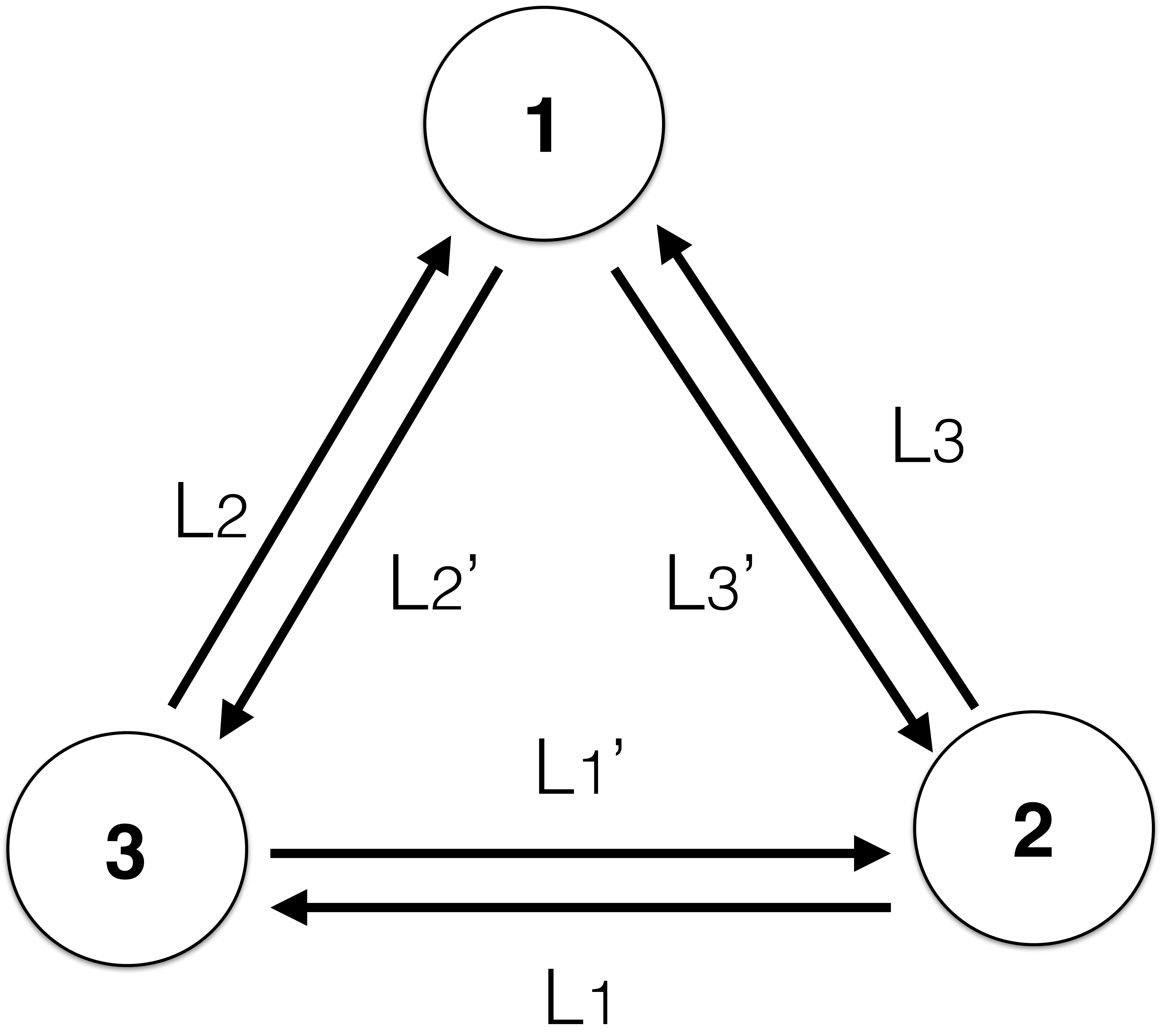
- Equal-arm interferometer detector
- Cancel laser frequency noise by comparing phases of split beams propagated along the equal arms of the detector.

Laser Interferometry



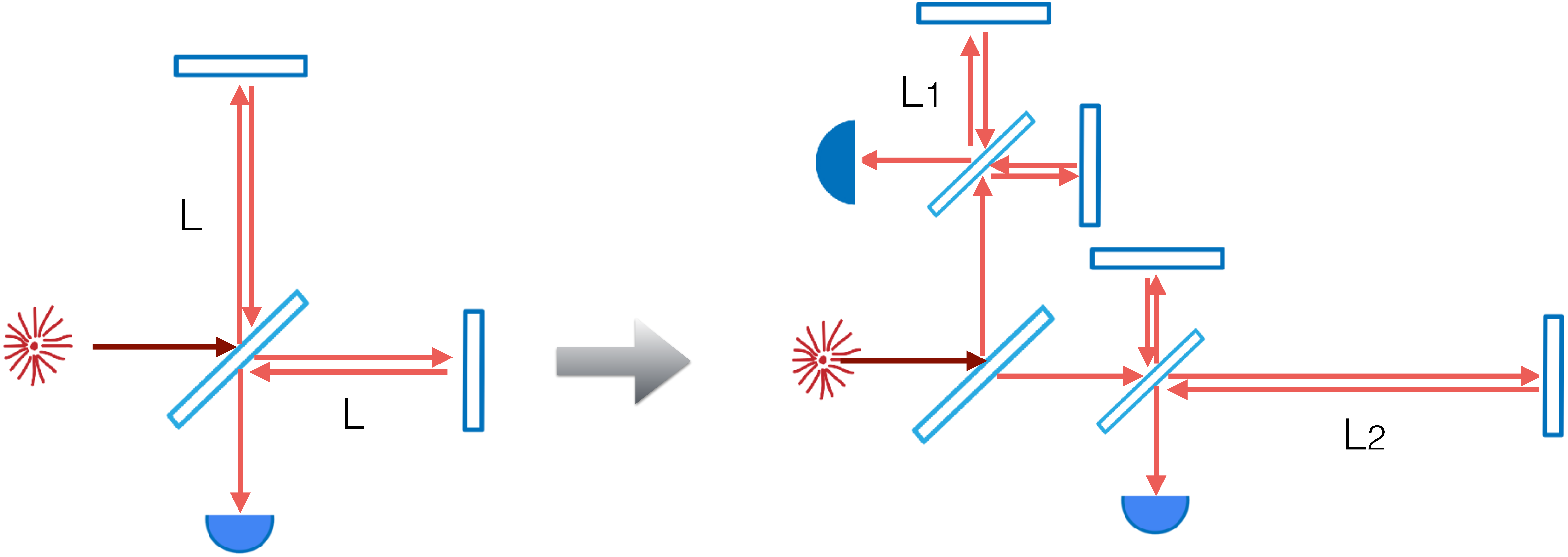
- Equal-arm interferometer detector
- Cancel laser frequency noise by comparing phases of split beams propagated along the equal arms of the detector.

Time Delay Interferometry



- Unequal arm interferometer
- How can we chancel laser noise?

Time Delay Interferometry



Unequal arm interferometer

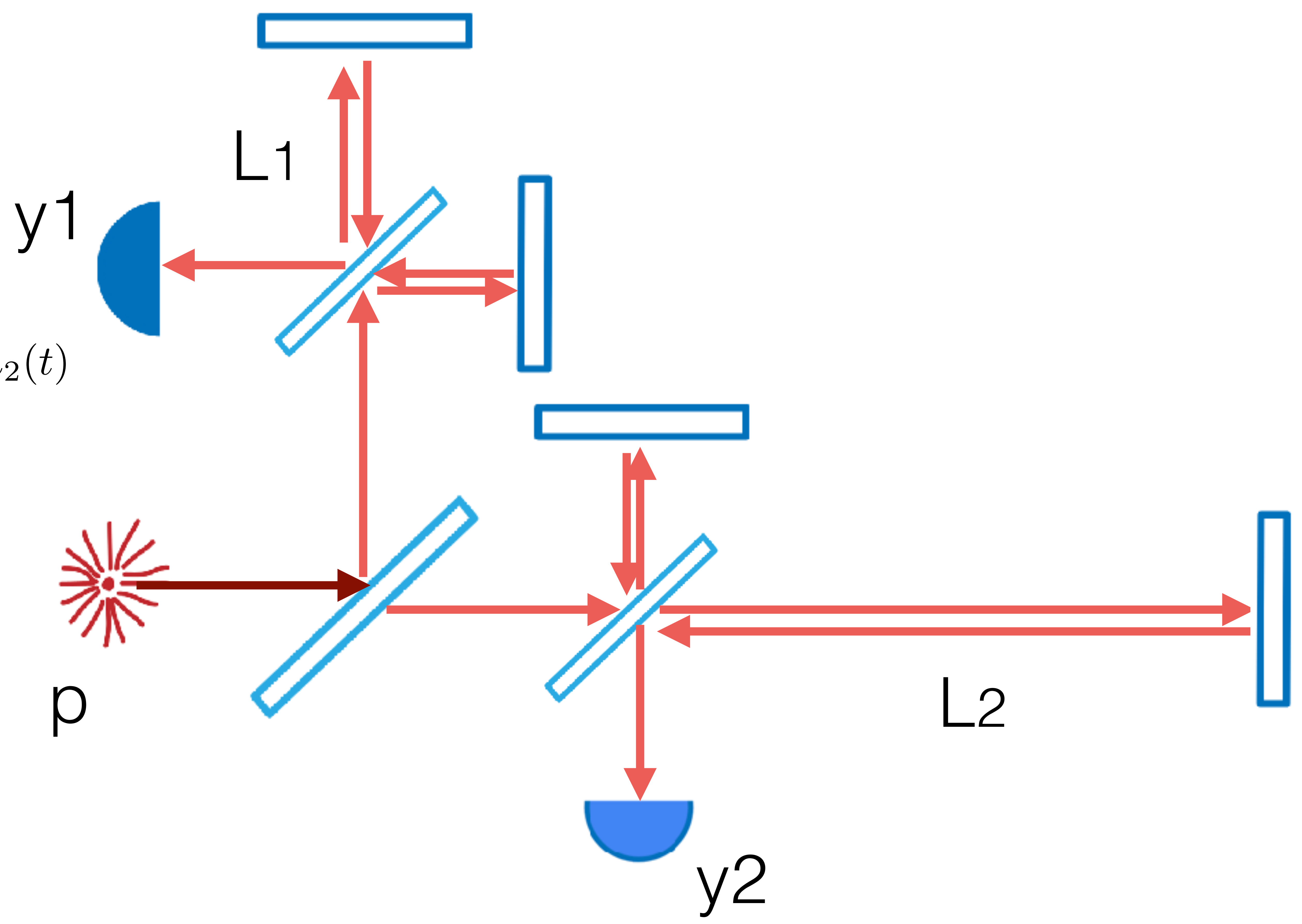
Time Delay Interferometry

- How can we chancel laser noise?

$$y_1(t) = p(t - 2L_1) - p(t) + h_1(t) + n_1(t)$$
$$y_2(t) = p(t - 2L_2) - p(t) + h_2(t) + n_2(t)$$

$$y_1(t) - y_2(t) = p(t - 2L_1) - p(t - 2L_2) + h_1(t) - h_2(t) + n_1(t) - n_2(t)$$

If we time-shift y1(t) by the round trip light time in arm L2
and subtract y2(t) time-shifted by the round trip in arm L1 we get:



Unequal arm interferometer

Time Delay Interferometry

$$y_1(t-2L_2) - y_2(t-2L_1) = p(t-2L_1) - p(t-2L_2) + h_1(t-2L_2) - h_2(t-2L_1) + n_1(t-2L_2) - n_2(t-2L_1)$$

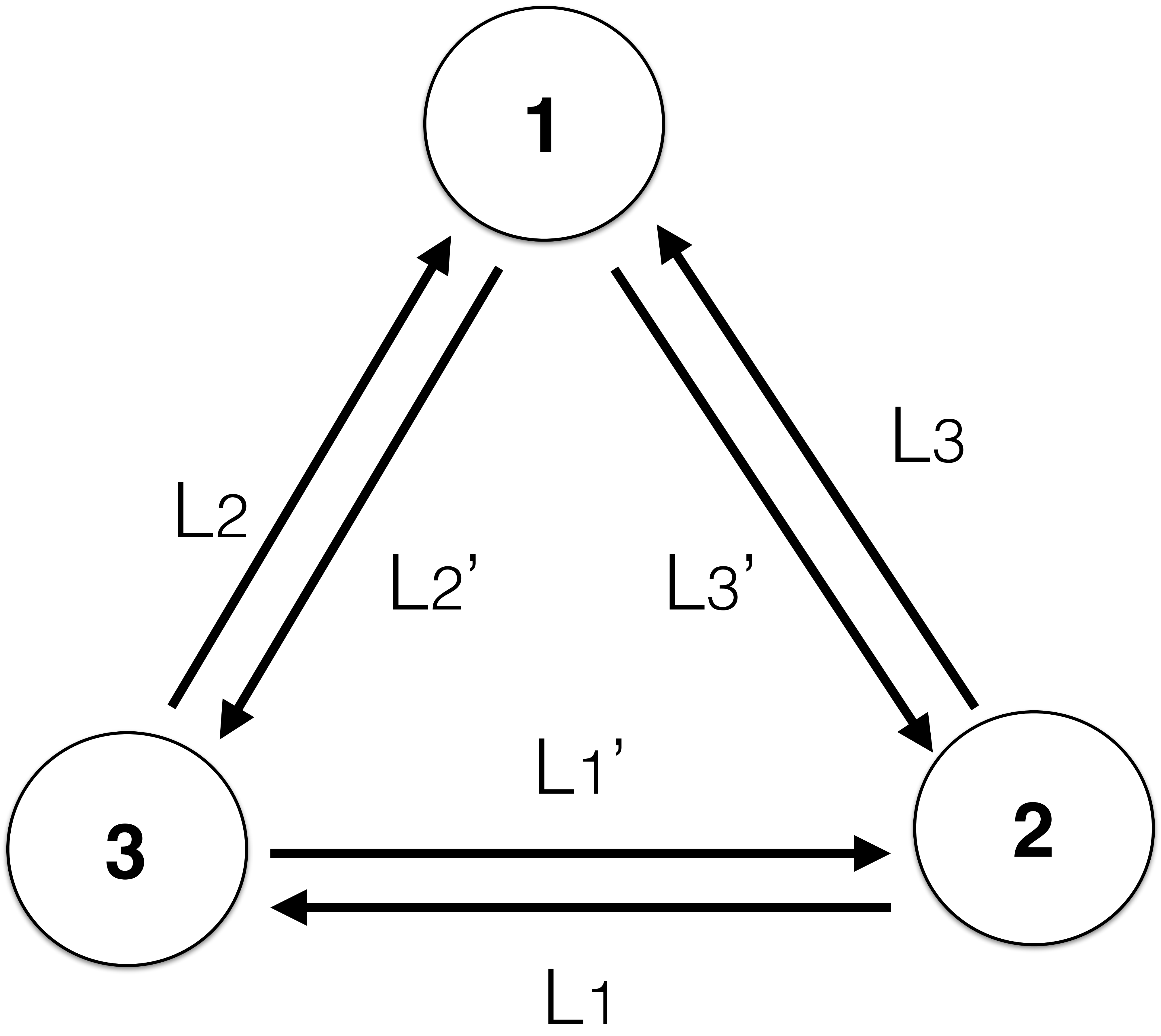
Combine it with previous expression

$$y_1(t) - y_2(t) = p(t-2L_1) - p(t-2L_2) + h_1(t) - h_2(t) + n_1(t) - n_2(t)$$

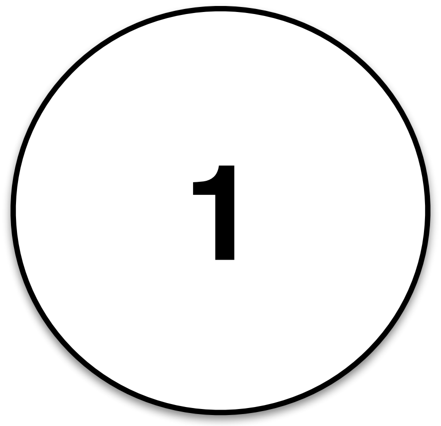
We get measurement free of laser noise

$$X = [y_1(t) - y_2(t)] - [y_1(t-2L_2) - y_2(t-2L_1)]$$

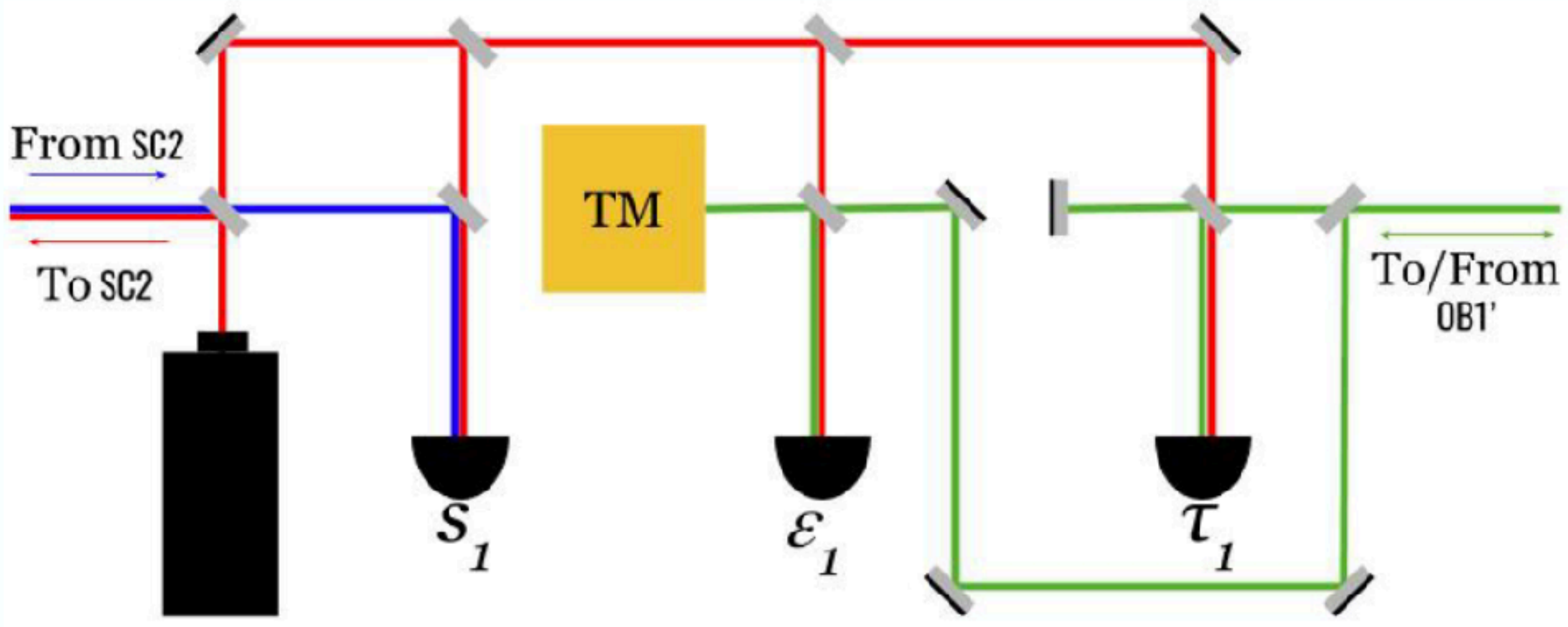
Ranging



Lets look inside one optical bench of



Ranging



LISA Optical Bench 1

s_1 — science interferometer

ε_1 — test mass interferometer

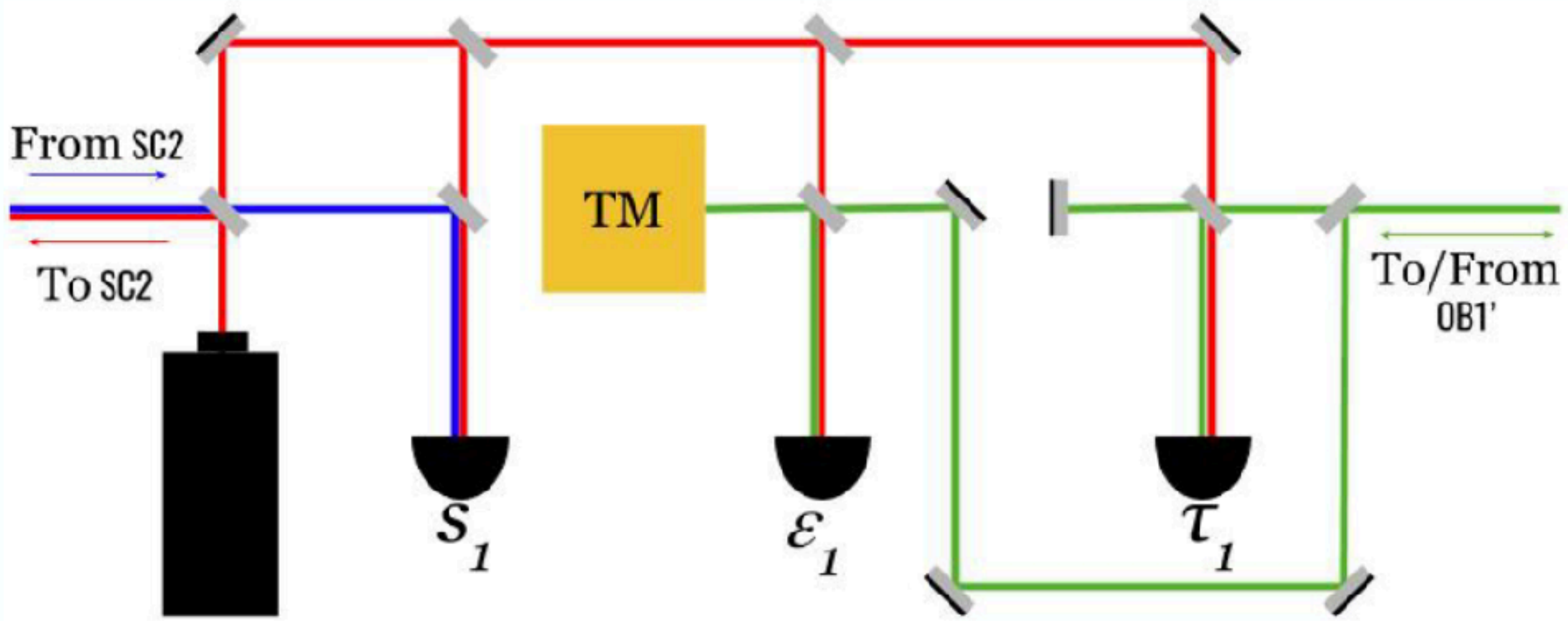
τ_1 — reference interferometer

$$s_1(t) = h_1(t) + p_{2'}(t - \frac{L_3(t)}{c}) - p_1(t) + \ldots$$

$$\varepsilon_1(t) = p_{1'}(t) - p_1(t) + \ldots$$

$$\tau_1(t) = p_{1'}(t) - p_1(t) + \ldots$$

Ranging



LISA Optical Bench 1

s_1 — science interferometer

ε_1 — test mass interferometer

τ_1 — reference interferometer

$$s_1(t) = h_1(t) + p_{2'}(t - \frac{L_3(t)}{c}) - p_1(t) + \dots$$

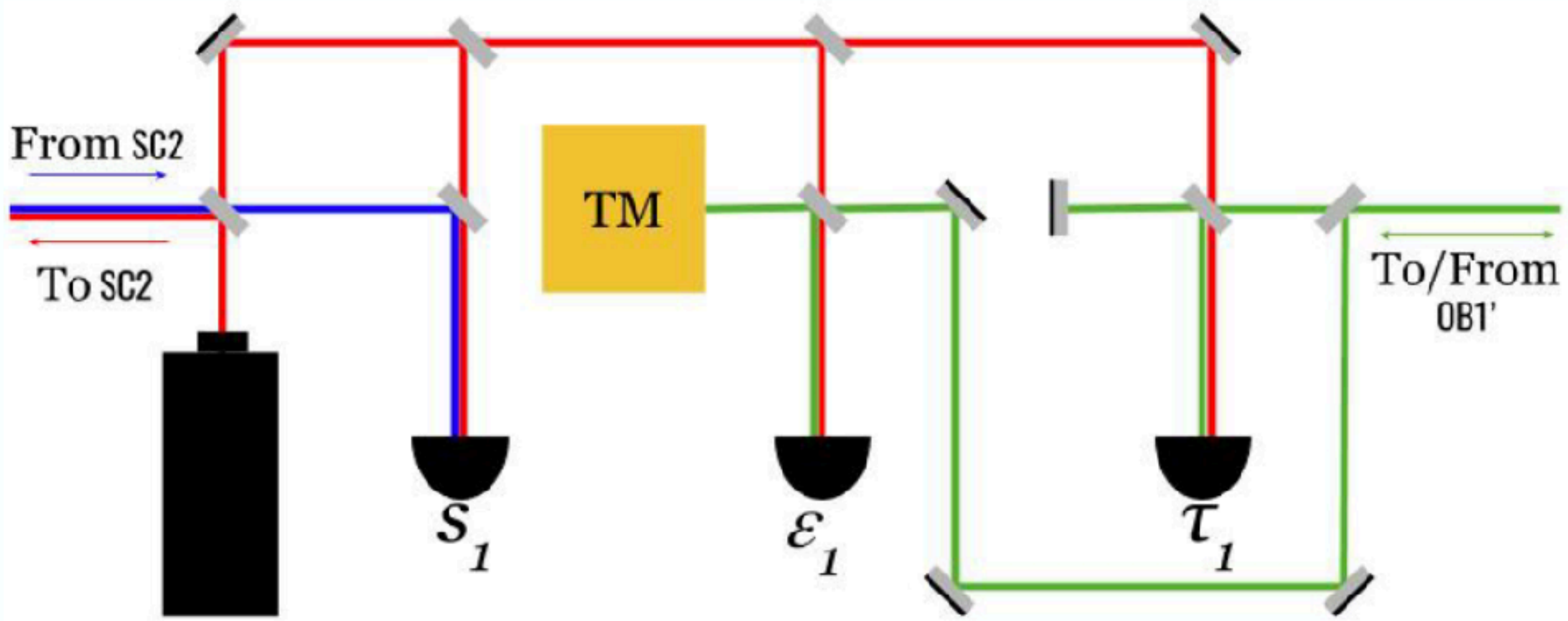
$$\varepsilon_1(t) = p_{1'}(t) - p_1(t) + \dots$$

$$\tau_1(t) = p_{1'}(t) - p_1(t) + \dots$$

Raw interferometer measurements

$$\left\{ \begin{array}{c} s_1,\ s_{1'},\ s_2,\ s_{2'},\ s_3,\ s_{3'}, \\ \varepsilon_1,\ \varepsilon_{1'},\ \varepsilon_2,\ \varepsilon_{2'},\ \varepsilon_3,\ \varepsilon_{3'}, \\ \tau_1,\ \tau_{1'},\ \tau_2,\ \tau_{2'},\ \tau_3,\ \tau_{3'} \end{array} \right\}$$

Ranging



LISA Optical Bench 1

s_1 — science interferometer

ε_1 — test mass interferometer

τ_1 — reference interferometer

$$s_1(t) = h_1(t) + p_{2'}(t - \frac{L_3(t)}{c}) - p_1(t) + \ldots$$

$$\varepsilon_1(t) = p_{1'}(t) - p_1(t) + \ldots$$

$$\tau_1(t) = p_{1'}(t) - p_1(t) + \ldots$$

Raw interferometer measurements

$$\left\{\begin{matrix} s_1, s_{1'}, s_2, s_{2'}, s_3, s_{3'}, \\ \varepsilon_1, \varepsilon_{1'}, \varepsilon_2, \varepsilon_{2'}, \varepsilon_3, \varepsilon_{3'}, \\ \tau_1, \tau_{1'}, \tau_2, \tau_{2'}, \tau_3, \tau_{3'} \end{matrix}\right\} \Rightarrow \{X, Y, Z\}$$

Michelson combinations

Time Delay Interferometry

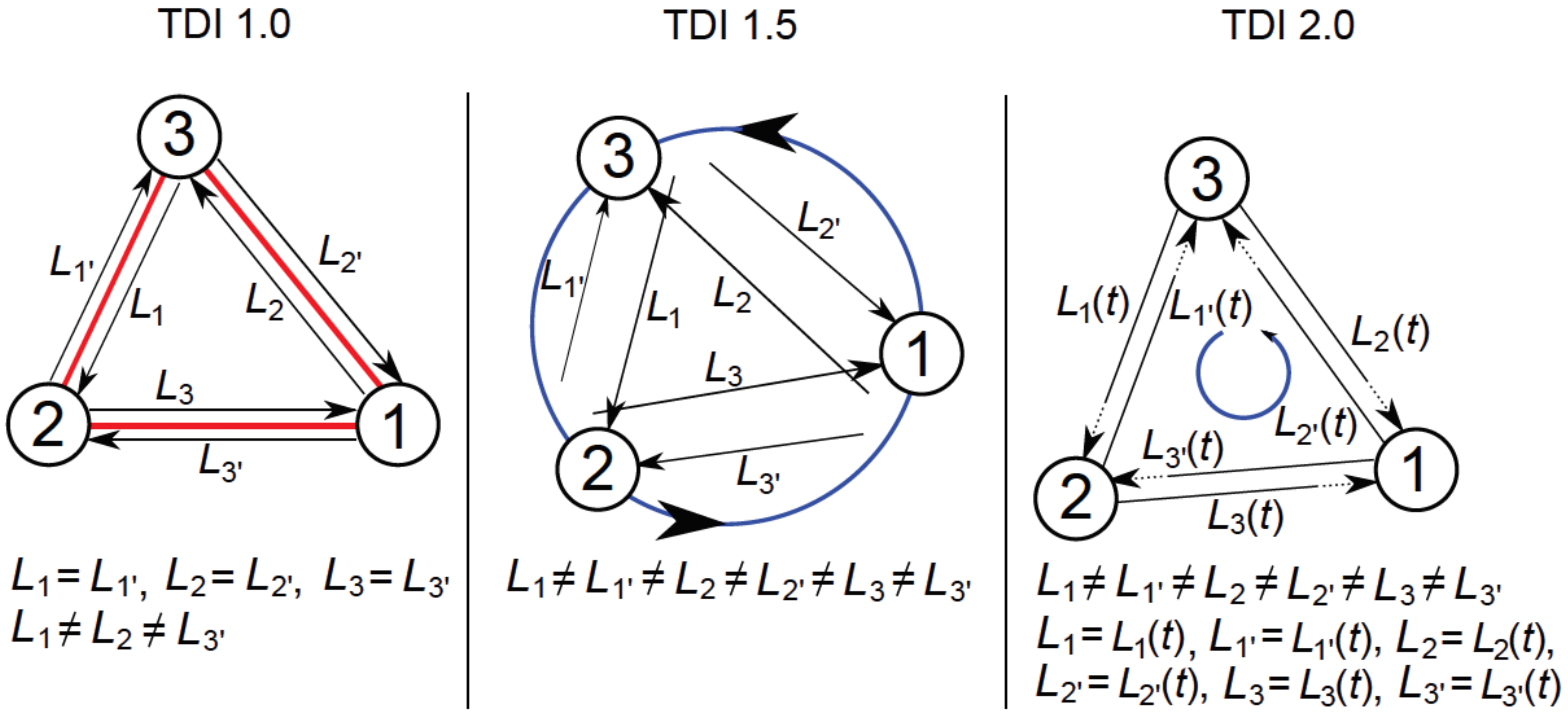


Image: Markus Otto thesis

TDI Ranging

- Performing TDI requires the measurement of the photon travel time!
- Can be performed with Pseudo Random Noise modulation scheme.
- TDI ranging (TDIR) is an alternative data driven method.
- TDIR — independent and unbiased method

  consistency check

  fallback

  can be performer directly with source parameters estimation

TDIR Bayesian way

Usually in gravitational wave data analysis we use Bayes' theorem

$$p(\theta|\mathbf{d}) = \frac{p(\mathbf{d}|\theta)p(\theta)}{p(\mathbf{d})}$$

If data can be represented as $\mathbf{d} = \mathbf{h}(\theta) + \mathbf{n}$

We can write the likelihood for the Gaussian noise

$$\mathcal{L}(\theta|\mathbf{d}) = \prod \frac{1}{\sqrt{(2\pi)^N|\mathbf{C}|}} \exp\left[-\frac{1}{2}(\mathbf{d} - \mathbf{h}(\theta))^T\mathbf{C}^{-1}(\mathbf{d} - \mathbf{h}(\theta))\right]$$

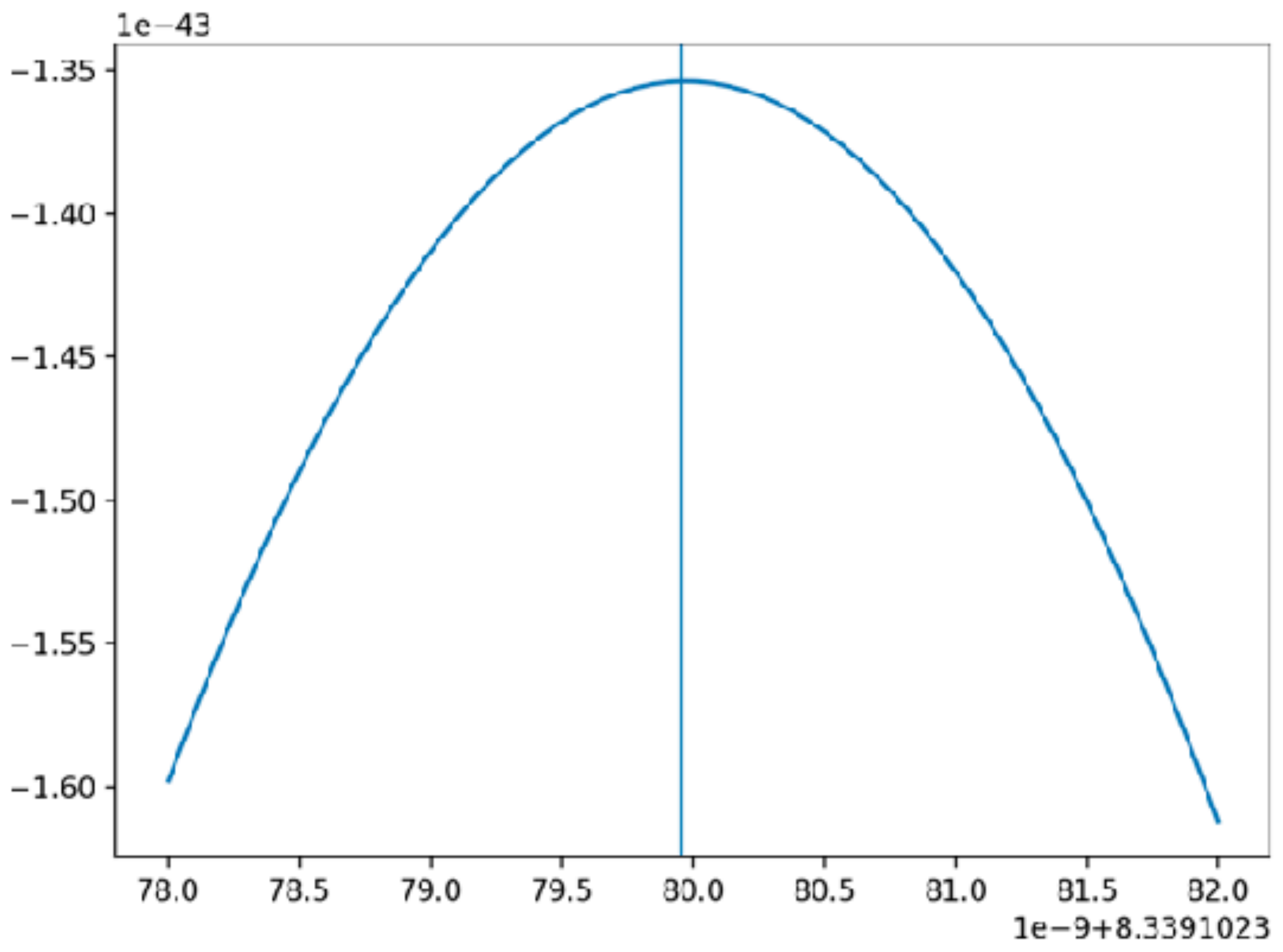
TDIR Bayesian way

- But in our case we cannot present data as a sum of the parameterised model and noise.

- Michelson combinations contain both noise and estimated parameter.

- We re-evaluate Michelson combinations on each iteration.

$$\log \mathcal{L} \propto -\sum \mathbf{X}^T \mathbf{C}^{-1} \mathbf{X}$$



TDIR Bayesian way

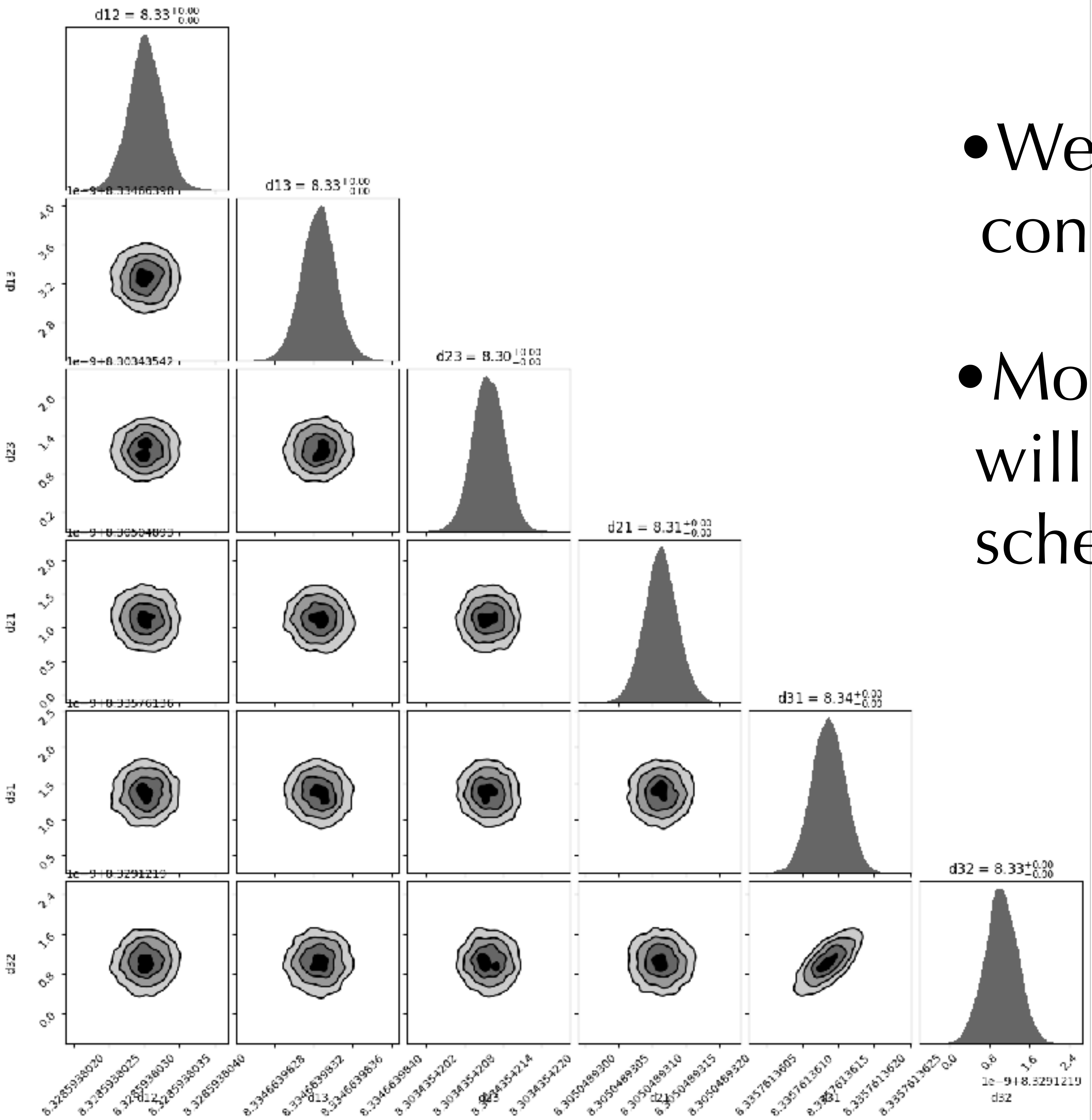
For this approach we have to define the model of the delay.

$$L_i = \text{const}$$

$$L_i(t) = L_i^0 + L_i^1 t$$

$$\ldots$$

TDIR Bayesian way

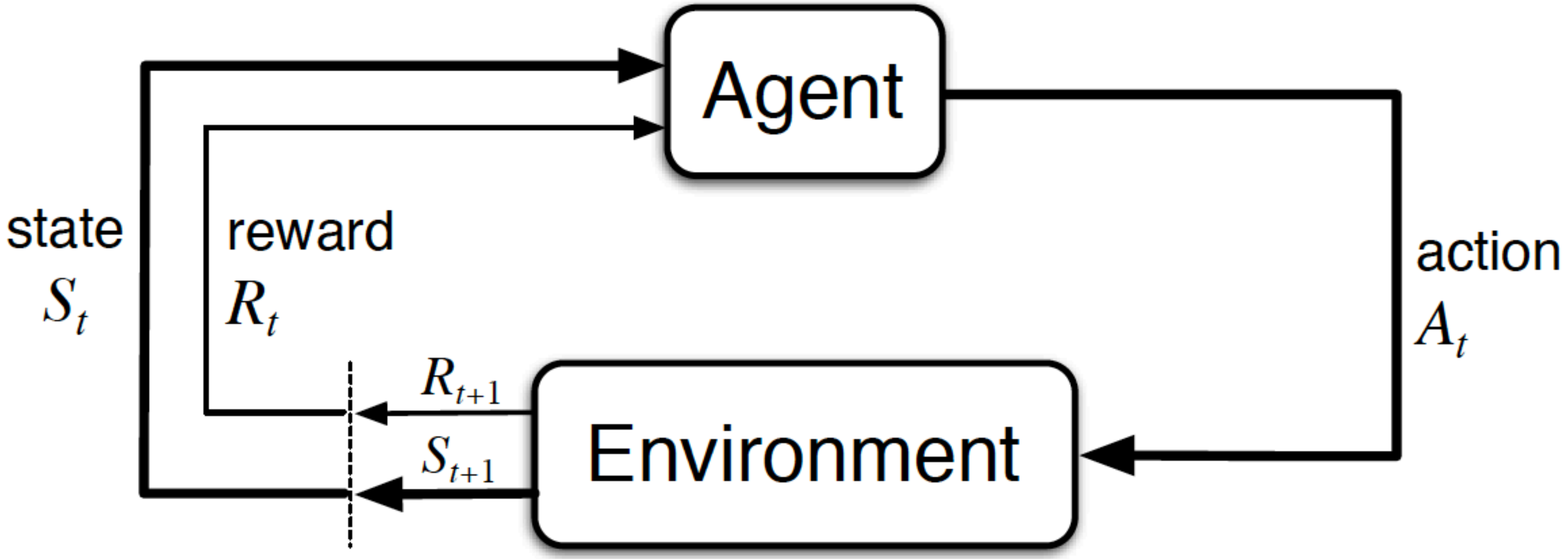


- We have performed Bayesian PE for constant unequal delays.
- More complicated model with this approach will require adjustment of the interpolation scheme.
- We will use this result as a benchmark to compare with the other method.

TDIR

- We want to investigate if it is possible to estimate model-free delays.
- Machine learning is usually a good approach for the data driven optimisation without model assumptions.
- In our case we do not have access to true values,
  but can have access directly to the cost function of the model
  which is represented with the likelihood
  and is evaluated outside of the optimisation algorithm.
- This means that we cannot use supervised (or unsupervised) learning but we have to use Reinforcement Learning for the optimisation.

TDIR Reinforcement Learning



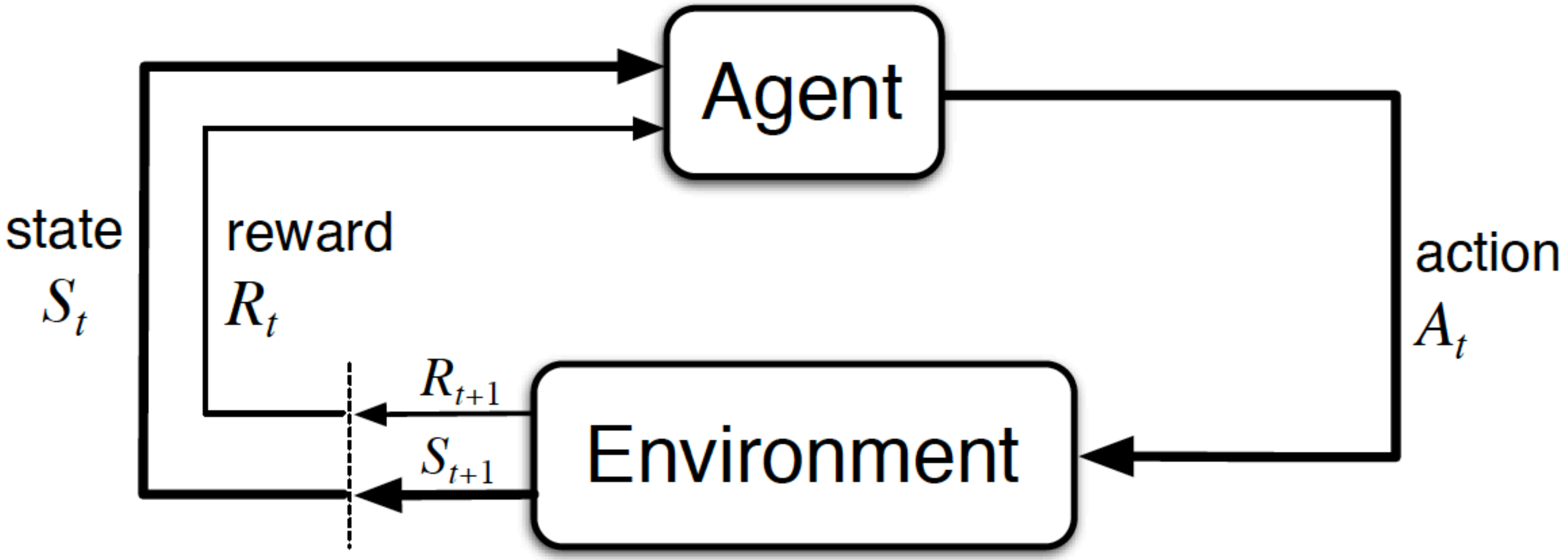
Agent interacts with an ***Environment***.
It *observes* the ***Environment*** and gets the ***States***.
The ***Environment*** also gives rise to the ***Rewards***, the values which ***Agent*** tries to maximise.

TDIR Reinforcement Learning

At each time step, the ***Agent*** implements a mapping from ***States*** to probabilities of selecting each possible ***Action***. This mapping is called the agent's ***policy***

$$\pi_t(a|s)$$

Reinforcement learning methods specify how the ***Agent*** changes its ***policy*** as a result of its experience.



TDIR Reinforcement Learning

Formalisation of the policy search problem.

- policy: $\pi_t(a|s)$
- trajectory τ_i : action/reward pairs $(a_0, r_0, a_1, r_1, \ldots)$
- corresponding return: $R(\tau) = \sum_{t=0}^{T} \gamma^t r_t$

We want to find the optimal policy. We use policy gradient methods.

This can be done by finding the optimum of the *performance:* $J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[R(\tau)]$

TDIR Reinforcement Learning

We use Soft Actor-Critic (SAC) method.

Policy gradient is Actor-Critic can be written as:

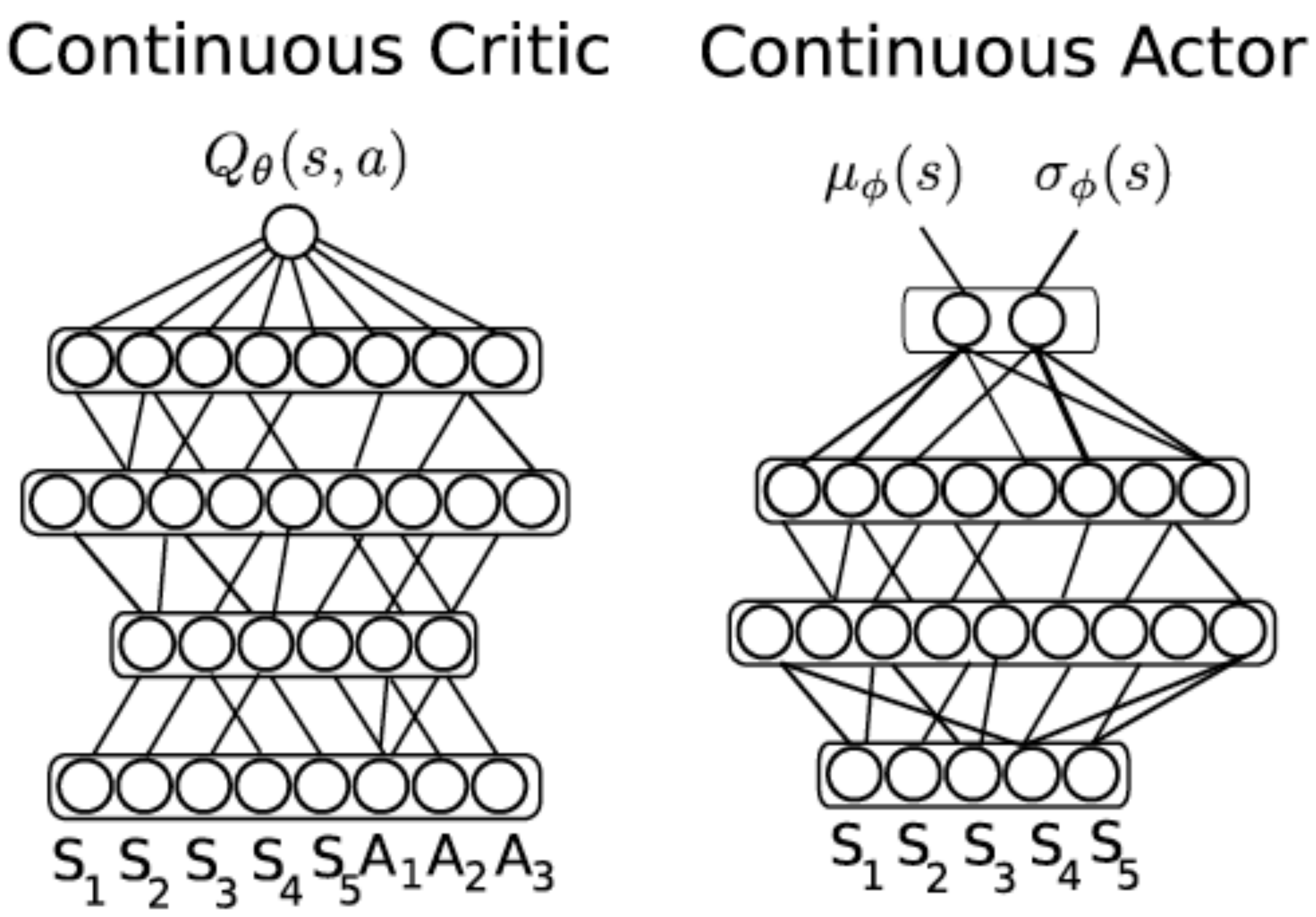
$$\nabla_\theta J(\theta) \approx \mathbb{E}_{\pi_\theta} \left[\nabla_\theta \log \pi_\theta(s, a) \; Q_w(s, a)\right]$$

$$\Delta\theta = \alpha \nabla_\theta \log \pi_\theta(s, a) \; Q_w(s, a)$$

$Q_w(s, a)$ is the action-value function,
that estimate how good it is for the agent to be in a given state.

TDIR Reinforcement Learning

The value function and the policy are parameterised by the neural networks:

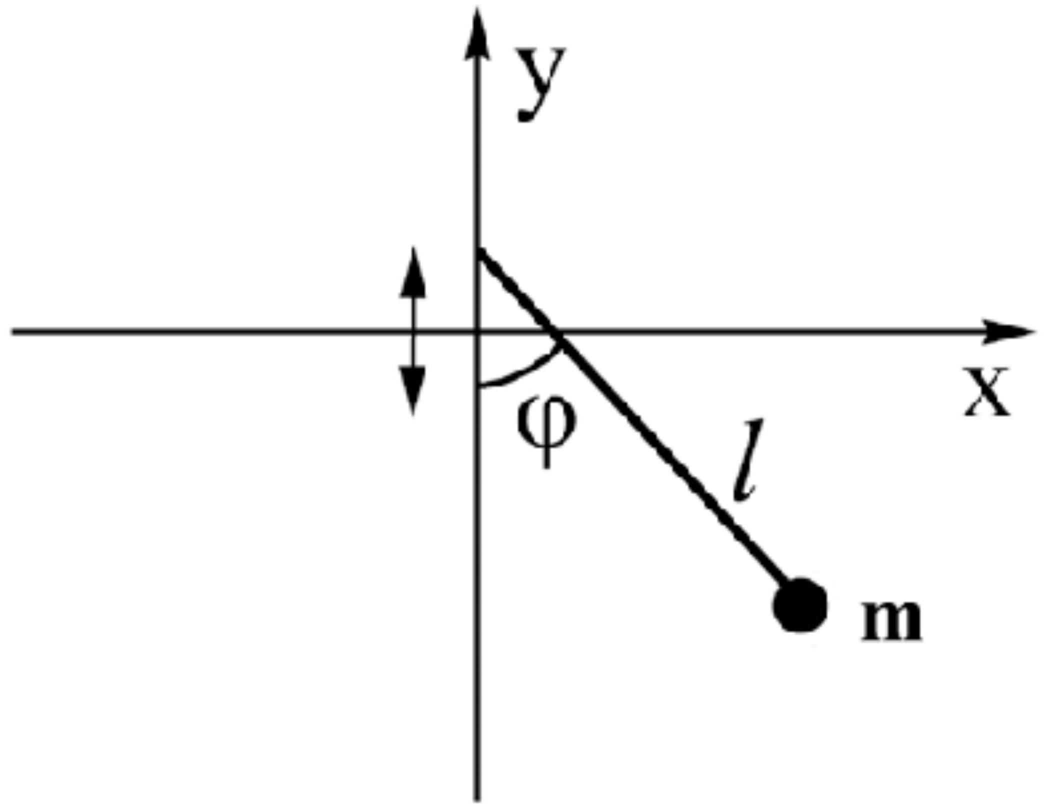


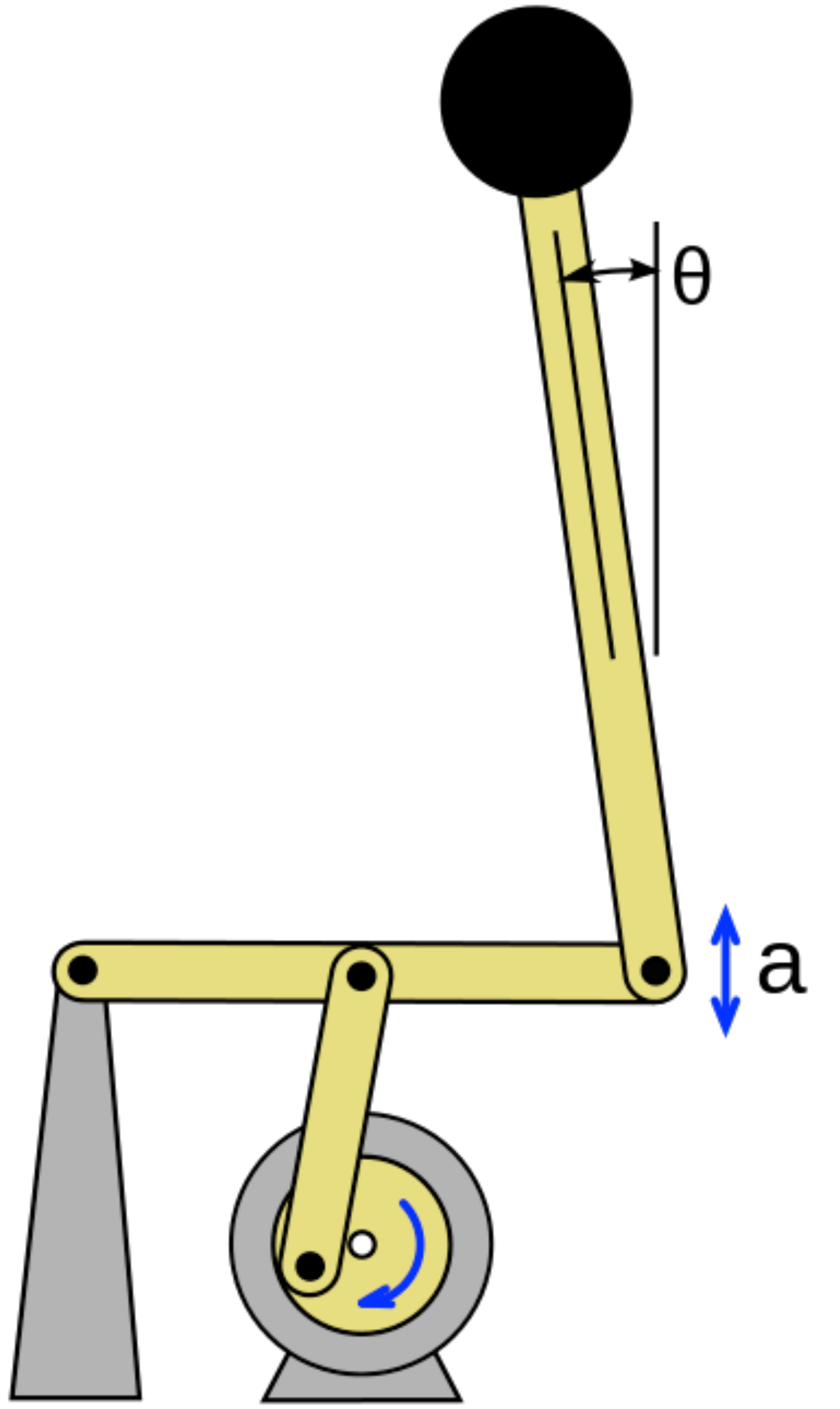
TDIR Reinforcement Learning

Environment: Inverted pendulum

$$\ddot{\varphi} = -(g + a\,\nu^2 \cos \nu t)\frac{\sin \varphi}{l}$$

- ν—frequency of the vertical oscillations of the suspension,
- a — amplitude of the oscillations of the suspension,
- $\omega_0 = \sqrt{g/l}$ — proper frequency of the mathematical pendulum,
- g — free fall acceleration,
- l — length of rigid and light pendulum,
- m — mass.





Observation: $(\cos(\varphi), \sin(\varphi), \dot{\varphi})$

State: $(\varphi, \dot{\varphi})$

Action: u torque

Reward: $-(||\varphi||^2 + 0.1\dot{\varphi} + 0.001u^2)$

State space equations:

$$\dot{\vec{x}} = A\vec{x} + Bu$$

$$\vec{y} = C\vec{x} + Du$$

TDIR Reinforcement Learning

Environment: Inverted pendulum

$$\ddot{\varphi} = -(g + a\,\nu^2 \cos \nu t)\frac{\sin \varphi}{l}$$

Observation: $(\cos(\varphi), \sin(\varphi), \dot{\varphi})$

State: $(\varphi, \dot{\varphi})$

Action: u torque

Reward: $-(||\varphi||^2 + 0.1\dot{\varphi} + 0.001u^2)$

Environment: LISA:

real physics (equations of motion + DFACS)
or at the moment a simulator

Observation: Values of the delays

||

State: Values of the delays

Action: Positive or negative increment to delay

Reward: Likelihood value

Results and Conclusions

- ❖ Implemented Bayesian approach for fixed arm-lengths.
- ❖ Investigate alternative interpolation schemes, to allow sampling for more complicated models.
- ❖ Reinforcement Learning approach is implemented.
- ❖ At the moment the method is very slow for unmodelled delays, work in progress to speed it up.
- ❖ Verify results of Reinforcement Learning against Bayesian parameter estimates.