Factor Graphs for Time Series Structured Optimization

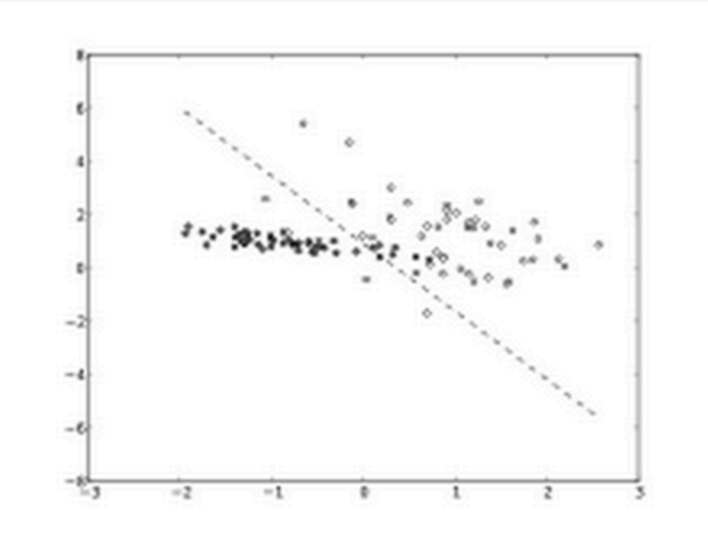
Class 23. 20 Nov 2014

Instructor: Gary Overett

Machine Learning Context

...according to infallible Wikipedia!

Machine learning and data mining



Problems

Classification · Clustering · Regression · Anomaly detection · Association rules · Reinforcement learning · Structured prediction · Feature learning · Online learning · Semi-supervised learning · Grammar induction

Supervised learning
(**classification** · **regression**)

Decision trees · Ensembles (Bagging, Boosting, Random forest) · *k*-NN · Linear regression · Naive Bayes · Neural networks · Logistic regression · Perceptron · Support vector machine (SVM) · Relevance vector machine (RVM)

Clustering

BIRCH · Hierarchical · *k*-means · Expectation-maximization (EM) · DBSCAN · OPTICS · Mean-shift

Dimensionality reduction

Factor analysis · CCA · ICA · LDA · NMF · PCA · t-SNE

Structured prediction

Graphical models (Bayes net, CRF, HMM)

Anomaly detection

k-NN · Local outlier factor

Neural nets

Autoencoder · Deep learning · Multilayer perceptron · RNN · Restricted Boltzmann machine · SOM · Convolutional neural network

Theory

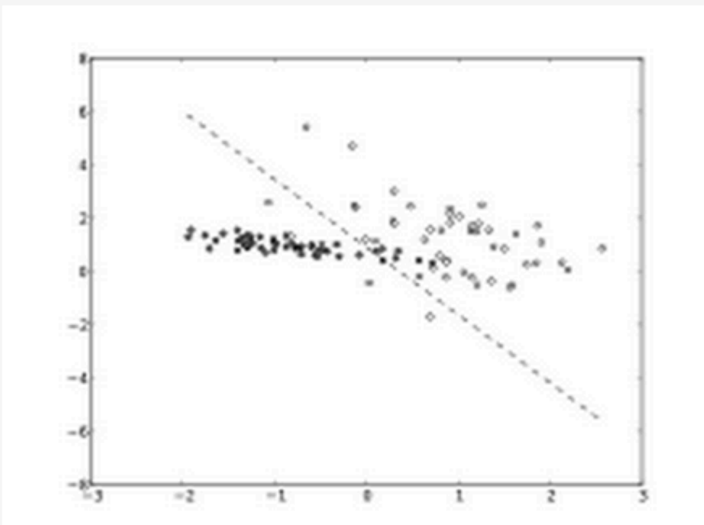
Bias-variance dilemma · Computational learning theory · Empirical risk minimization · PAC learning · Statistical learning · VC theory

Source: http://en.wikipedia.org/wiki/Machine_learning

Machine Learning Context

…according to infallible Wikipedia!

Machine learning and data mining



Problems

Classification · Clustering · Regression · Anomaly detection · Association rules · Reinforcement learning · Structured prediction · Feature learning · Online learning · Semi-supervised learning · Grammar induction

Supervised learning
(classification · regression)

Decision trees · Ensembles (Bagging, Boosting, Random forest) · *k*-NN · Linear regression · Naive Bayes · Neural networks · Logistic regression · Perceptron · Support vector machine (SVM) · Relevance vector machine (RVM)

Clustering

BIRCH · Hierarchical · *k*-means · Expectation-maximization (EM) · DBSCAN · OPTICS · Mean-shift

Dimensionality reduction

Factor analysis · CCA · ICA · LDA · NMF · PCA · t-SNE

Structured prediction

Graphical models (Bayes net, CRF, HMM)

Anomaly detection

k-NN · Local outlier factor

Neural nets

Autoencoder · Deep learning · Multilayer perceptron · RNN · Restricted Boltzmann machine · SOM · Convolutional neural network

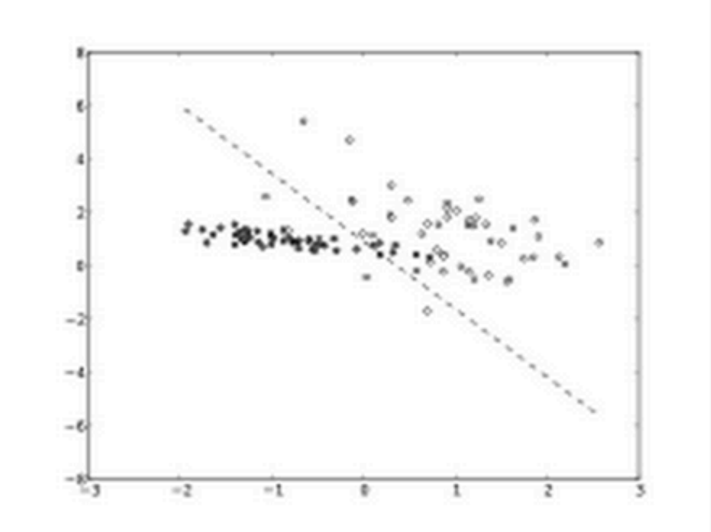
Theory

Bias-variance dilemma · Computational learning theory · Empirical risk minimization · PAC learning · Statistical learning · VC theory

Source: http://en.wikipedia.org/wiki/Machine_learning

Where we are exploring today?

Machine learning and data mining



Problems

Classification · Clustering · Regression · Anomaly detection · Association rules · Reinforcement learning · Structured prediction · Feature learning · Online learning · Semi-supervised learning · Grammar induction

Supervised learning
(**classification** · **regression**)

Decision trees · Ensembles (Bagging, Boosting, Random forest) · *k*-NN · Linear regression · Naive Bayes · Neural networks · Logistic regression · Perceptron · Support vector machine (SVM) · Relevance vector machine (RVM)

Clustering

BIRCH · Hierarchical · *k*-means · Expectation-maximization (EM) · DBSCAN · OPTICS · Mean-shift

Dimensionality reduction

Factor analysis · CCA · ICA · LDA · NMF · PCA · t-SNE

Structured prediction

Graphical models (Bayes net, CRF, HMM)

Anomaly detection

k-NN · Local outlier factor

Neural nets

Autoencoder · Deep learning · Multilayer perceptron · RNN · Restricted Boltzmann machine · SOM · Convolutional neural network

Theory

Bias-variance dilemma · Computational learning theory · Empirical risk minimization · PAC learning · Statistical learning · VC theory

Source: http://en.wikipedia.org/wiki/Machine_learning

What might you know about your "signal(s)"?

- Gaussian Distribution
- Made up of Independent Components
- Face-like
- Clustered
- Sparse
- Locally Smooth (or should be!)

What might you know about your "signal(s)"?

- Gaussian Distribution
- Made up of Independent Components
- Face-like
- Clustered
- Sparse
- Locally Smooth (or should be!)
- Choose or derive your constraints!

The more I know about my signal (or the signal I am trying to estimate) the more constraints I'm able to reasonably apply!

General vs. Domain Specific Constraints

General

- Gaussian
- Independent
- Smooth
- Non-Negative
- Sparse

Domain Specific

- N-Gram Speech Model (cat-sat-on-the-?->mat)
- Connectedness (Hand-connected-to-arm etc.)
- Ballistic Trajectory Prediction
- Vehicle Motion Model (cars-go-forward/ backwards not sideways)

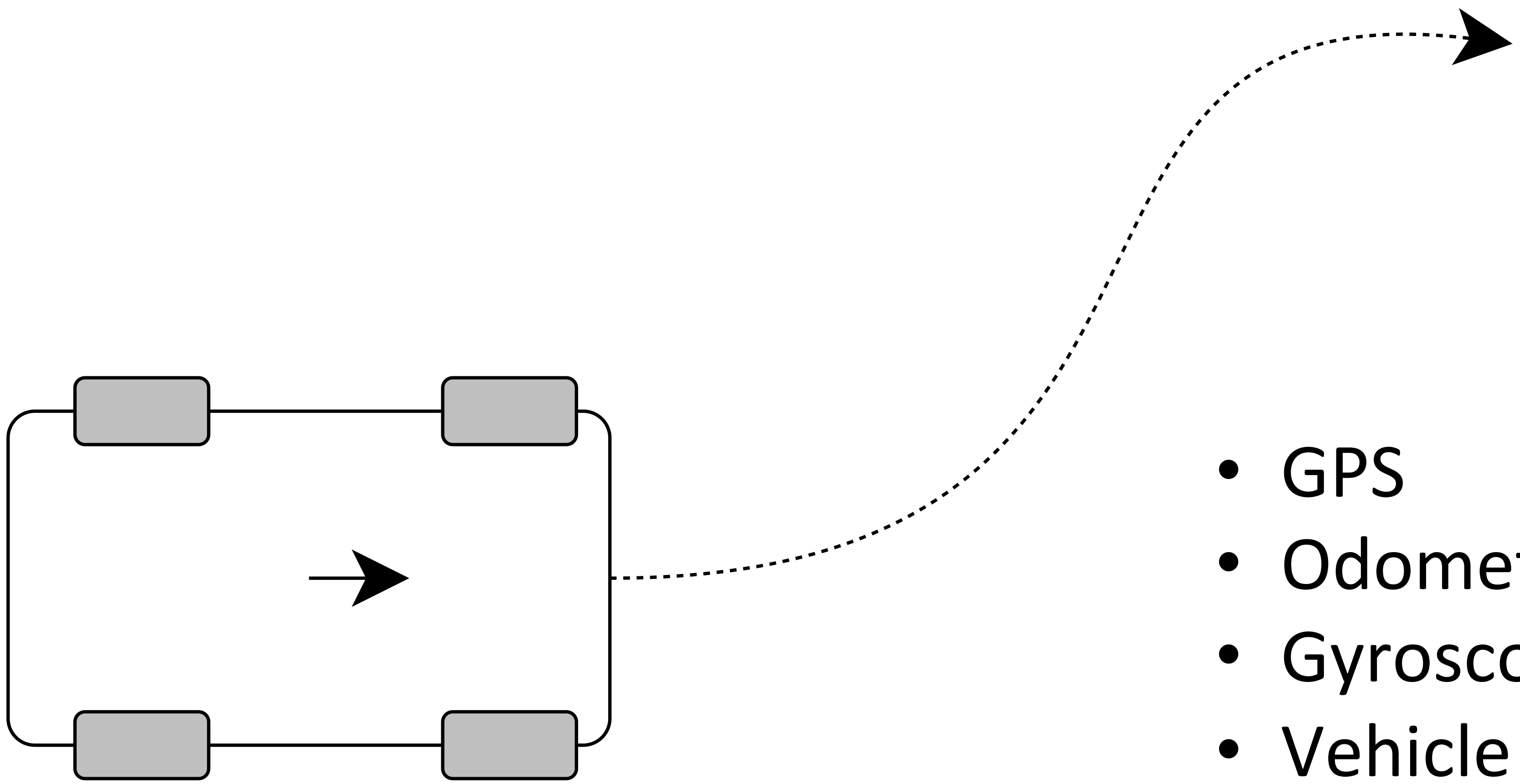
Consider the following 'signal'

We want to know the position of a survey vehicle in the real world.

Given the following time-series information:

- GPS location data at 1Hz (may suffer dropout)
- Wheel Encoder Odometry Measurements every 2m
- Inertial Sensor Measurements (Gyroscope) every 2m
- Basic Vehicle Geometry (wheel base width etc.)

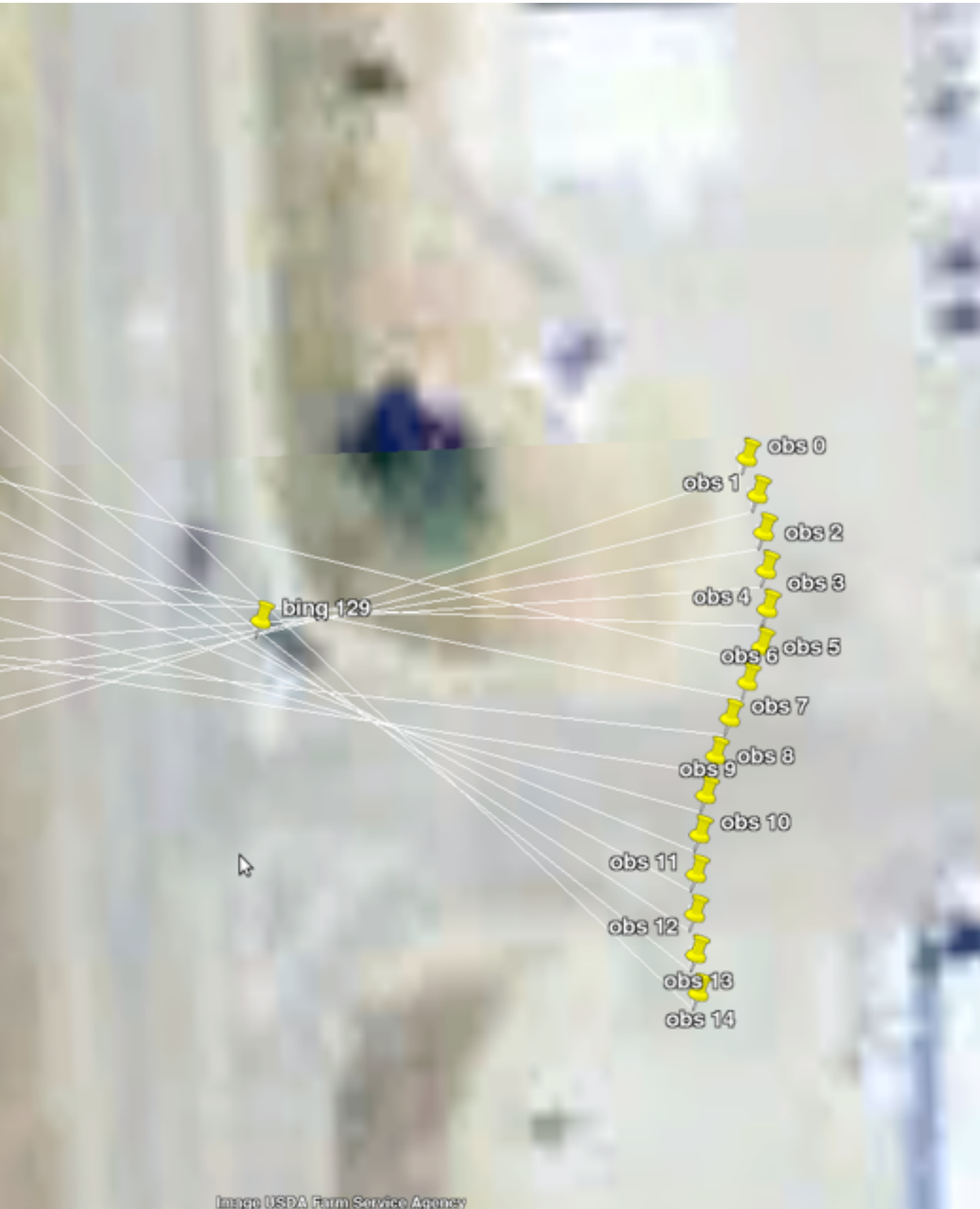
Vehicle Localization (Pose) Estimation

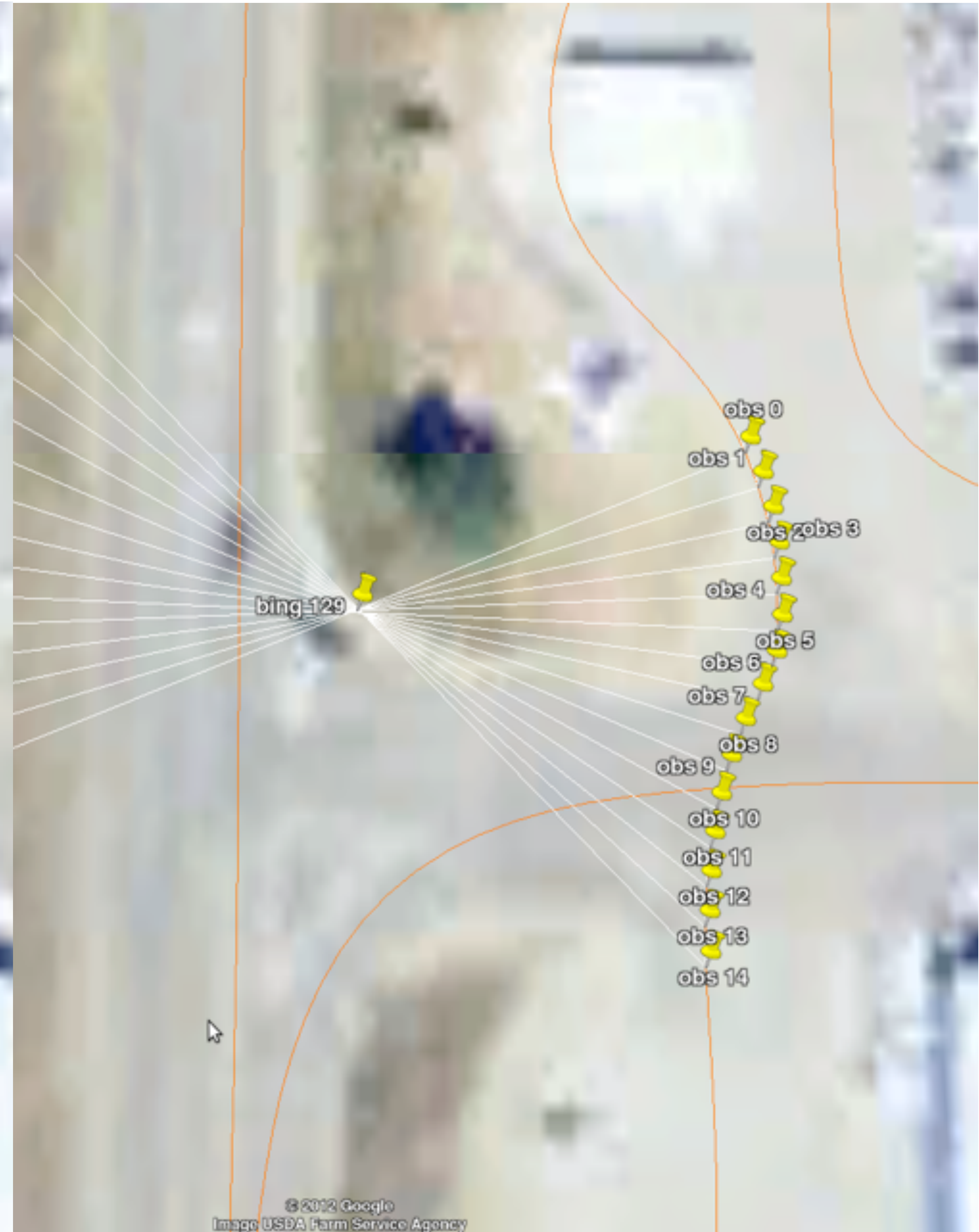


$$X = (x, y, \theta)$$

- GPS
- Odometry
- Gyroscope
- Vehicle Motion Constraints

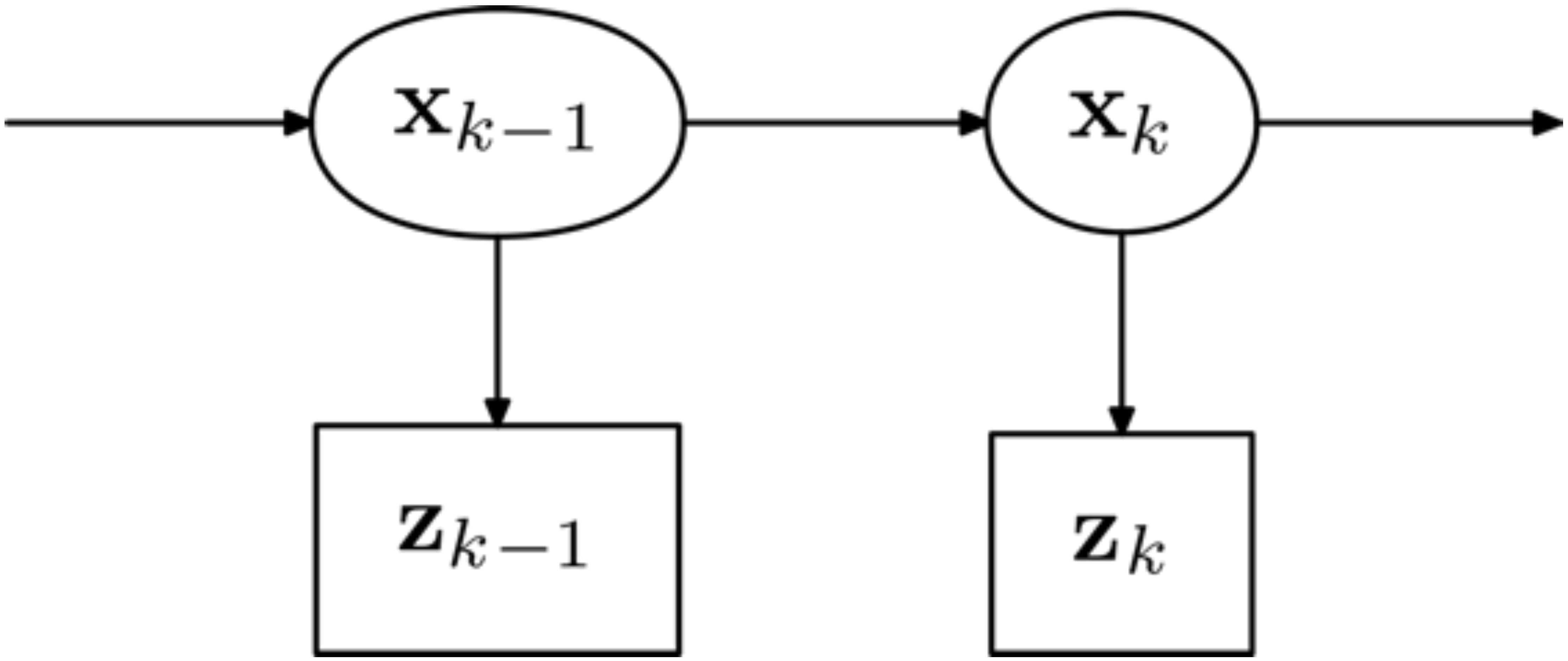
Use case : Vehicle/Camera Pose for Asset Localization



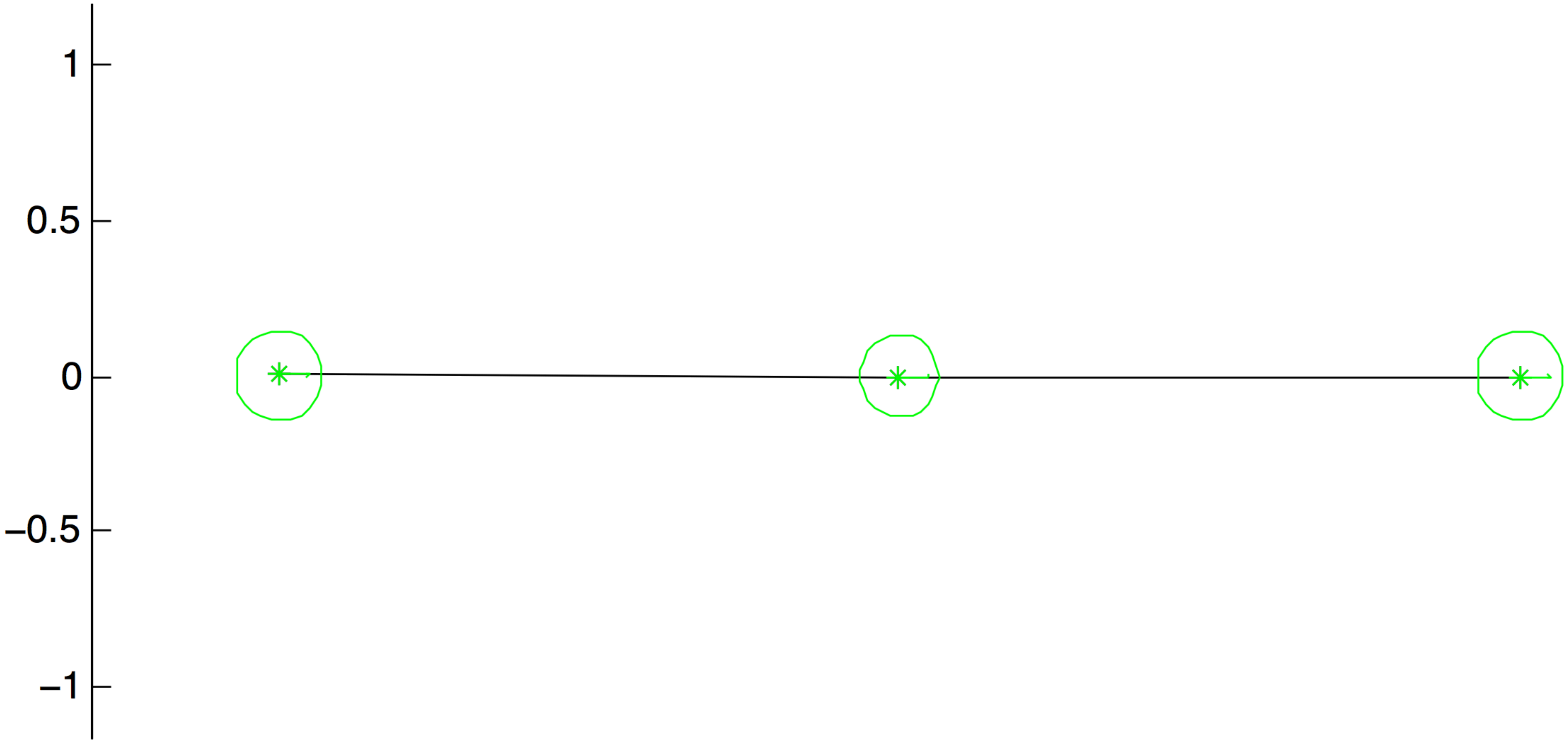


Kalman Filter : Fail!

Kalman Filter

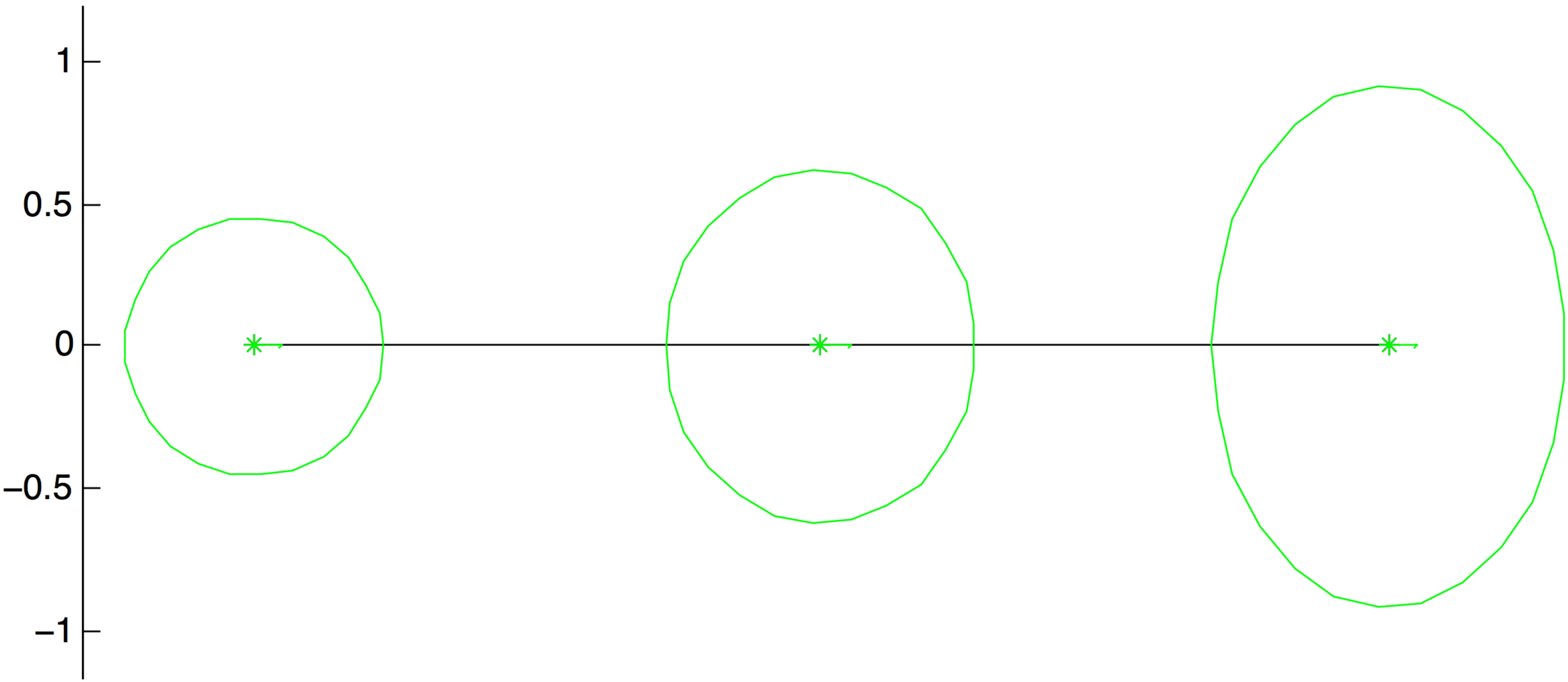


Certainty Over Time (Odo+GPS)



Credit: "Factor Graphs and GTSAM: A hands on introduction", Frank Dellaert

Certainty Over Time (Odo only)



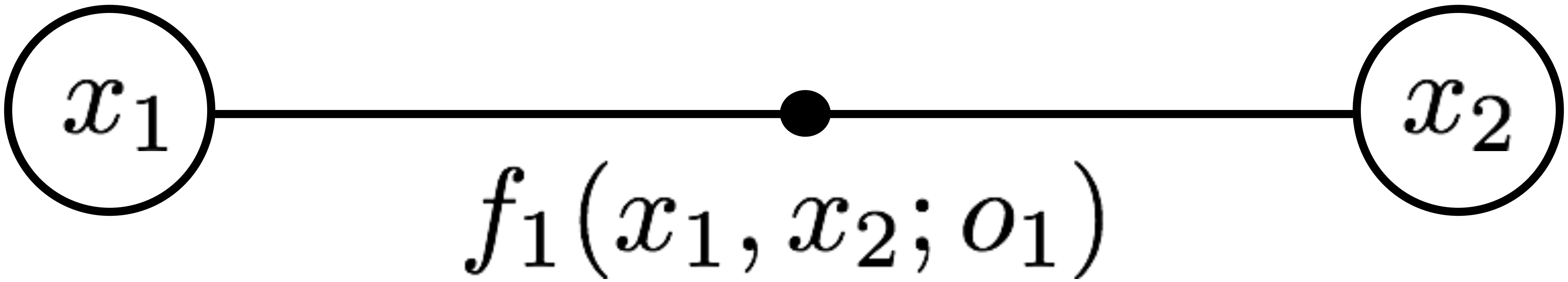
Credit: “Factor Graphs and GTSAM: A hands on introduction”, Frank Dellaert

Kalman Filter : Fail!

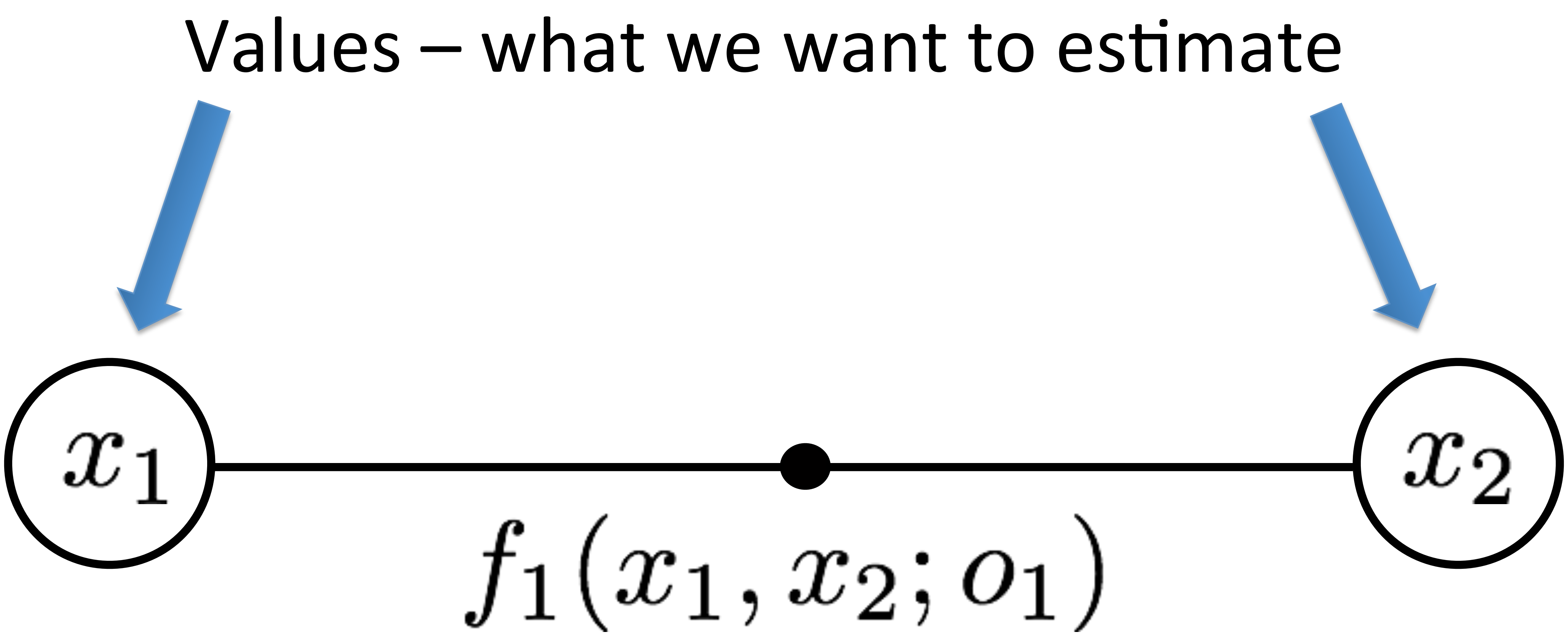
Survey Vehicle : Signal Behavior

- GPS Suffers Dropout, Noise and Discontinuities
- Odometry is very robust but multiple paths can explain a given odometry
- Gyroscope is has a high precision but ‘drifts’
- Vehicles travel along relatively smooth paths. Normal operation excludes lateral (sideways) motion.
- These constraints suggest a structured (graphical) prediction approach.

Structured Prediction : Factor Graphs



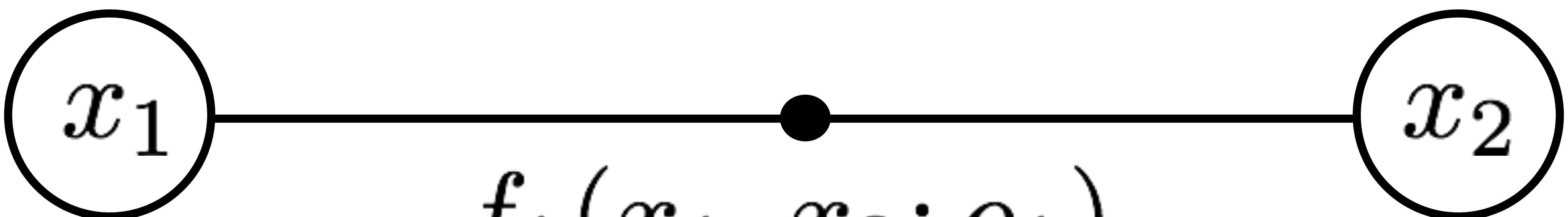
Factor Graphs



Factor Graphs

Values – what we want to estimate

- Vehicle Pose
- Error Corrected HDD Reader Output
- Image Segmentation Label
- Price of Apple Stock at time t etc.

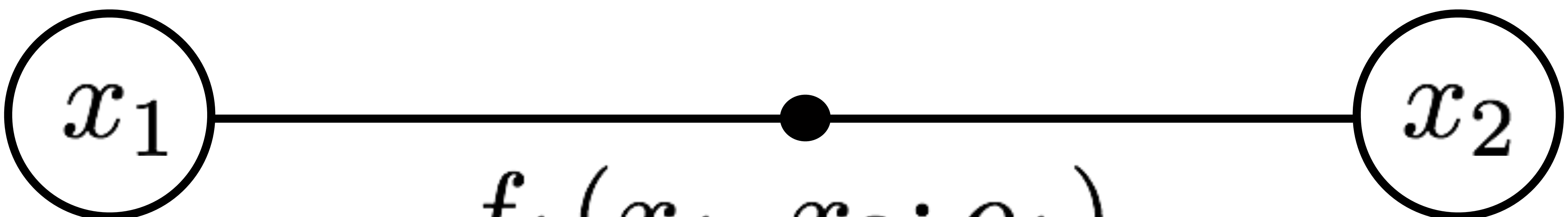


$$f_1(x_1, x_2; o_1)$$

Factor Graphs

Values – what we want to estimate

- Vehicle Pose
- Error Corrected HDD Reader Output
- Image Segmentation Label
- Price of Apple Stock at time t etc.



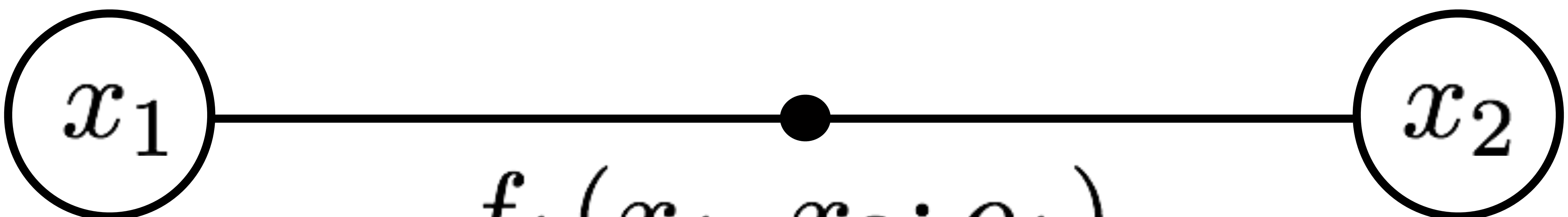
$$f_1(x_1, x_2; o_1)$$

Factor(s) – a constraint we formulate

Factor Graphs

Values – what we want to estimate

- Vehicle Pose
- Error Corrected HDD Reader Output
- Image Segmentation Label
- Price of Apple Stock at time t etc.

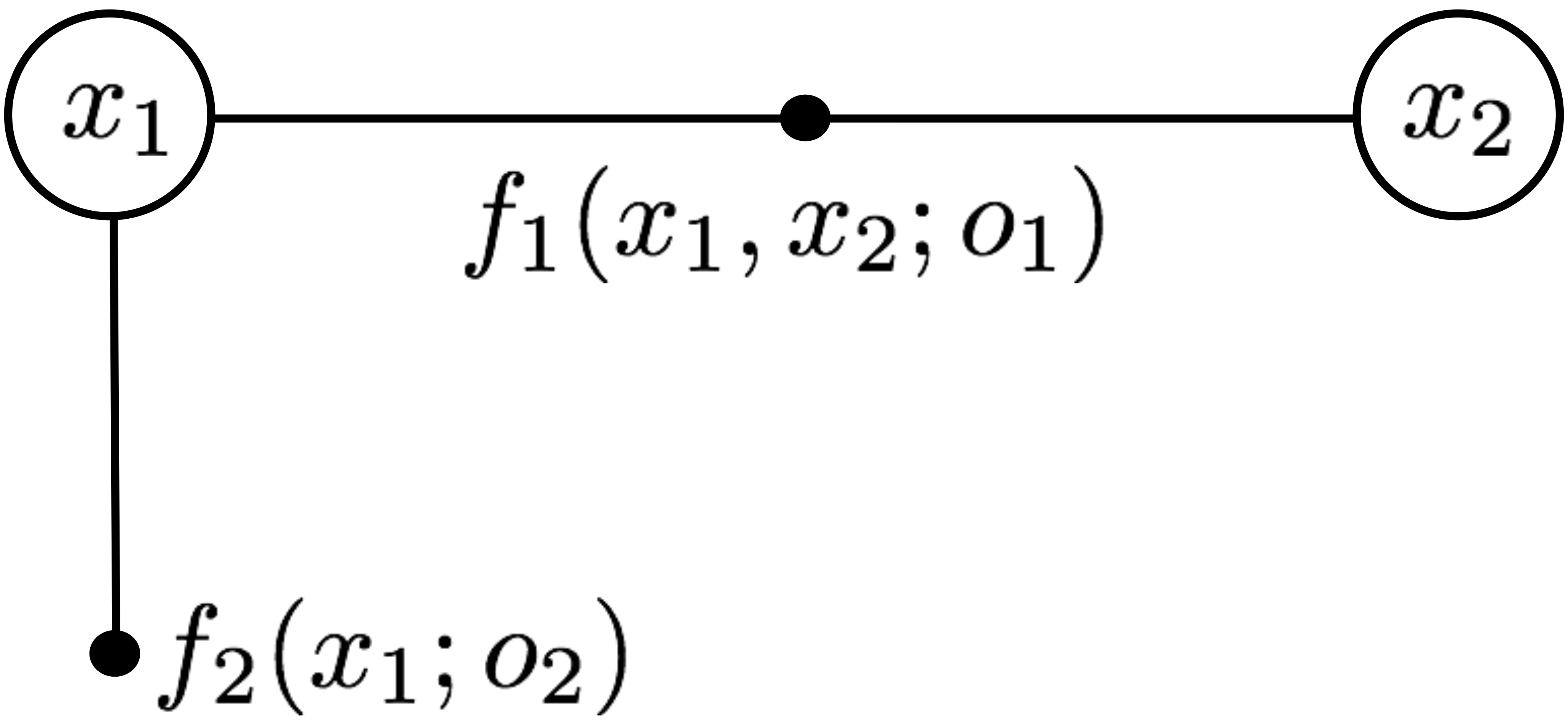


$$f_1(x_1, x_2; o_1)$$

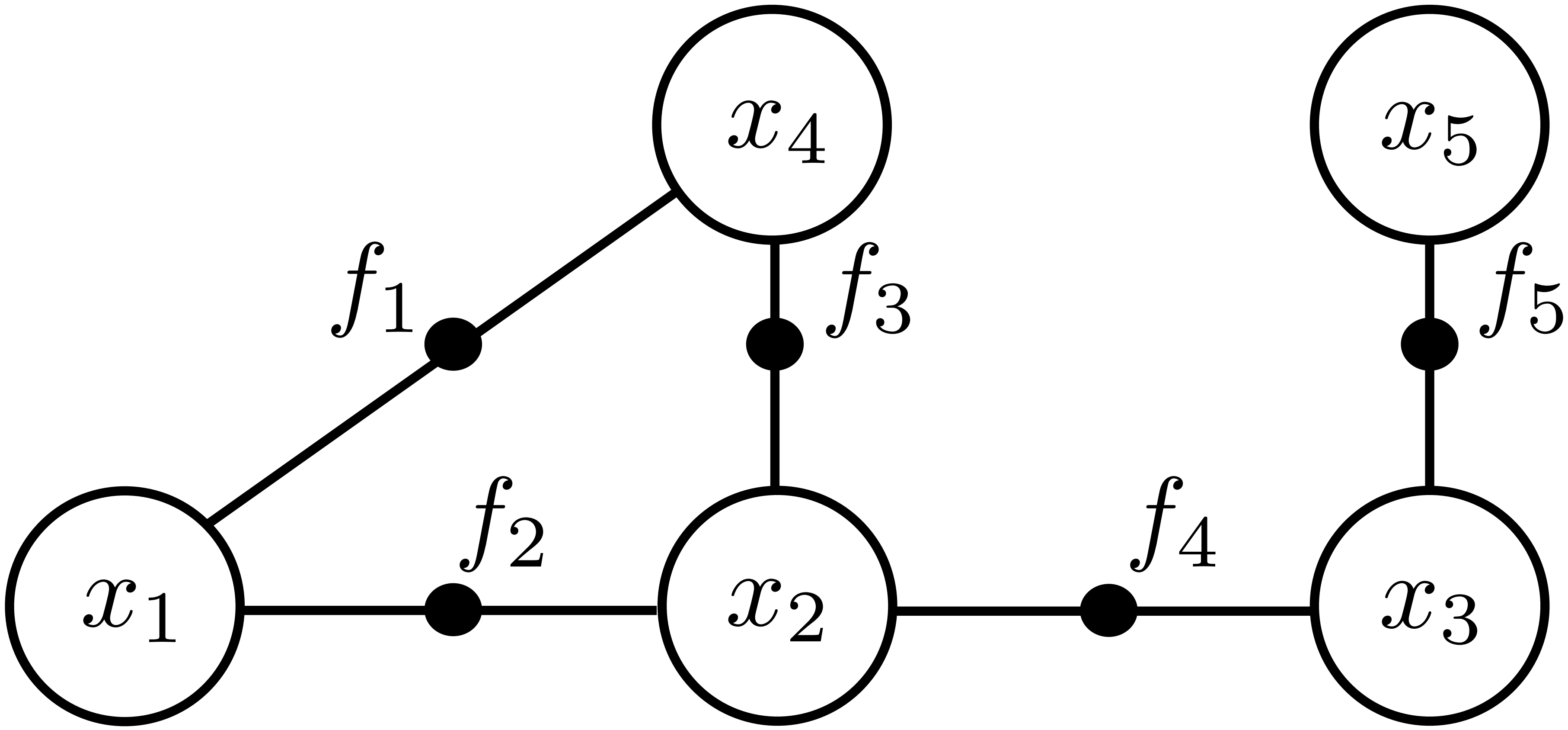
Factor(s) – a constraint we formulate

- Odometry Constraint
- Error Correcting Code
- Segmentation Prior (Green -> Trees)
- Todays stock price same as yesterday

Factor Graphs

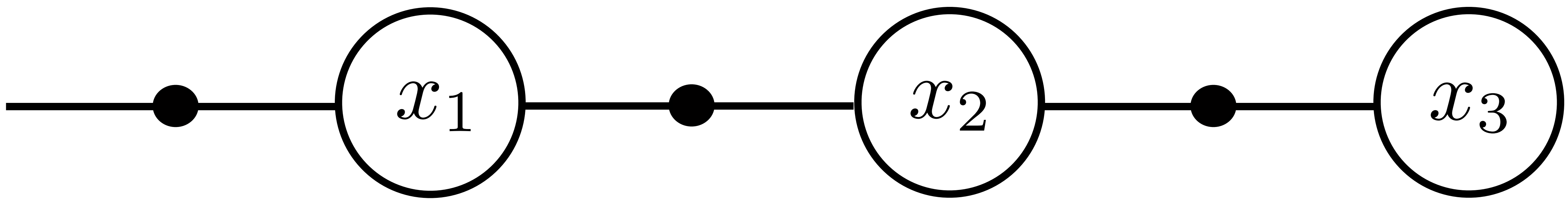


Factorization



$$g(x_1, x_2, x_3, x_4, x_5) = f_1(x_1, x_4) f_2(x_1, x_1) f_3(x_2, x_4) f_4(x_2, x_3) f_5(x_3, x_5)$$

Example Factorization : Markov Chain



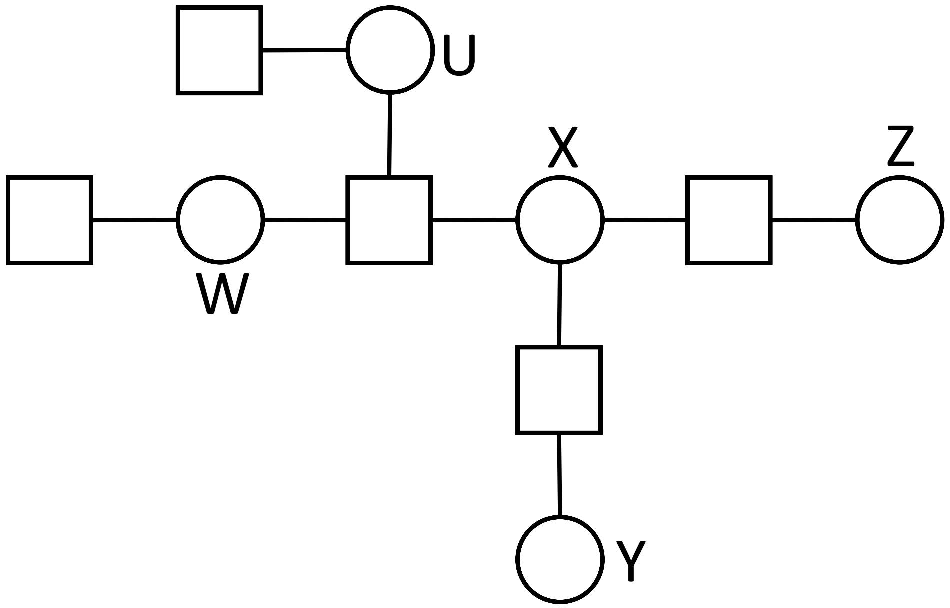
$$p(x_1, x_2, x_3) = p(x_1)p(x_2|x_1)p(x_3|x_2)$$

Unifying View of…

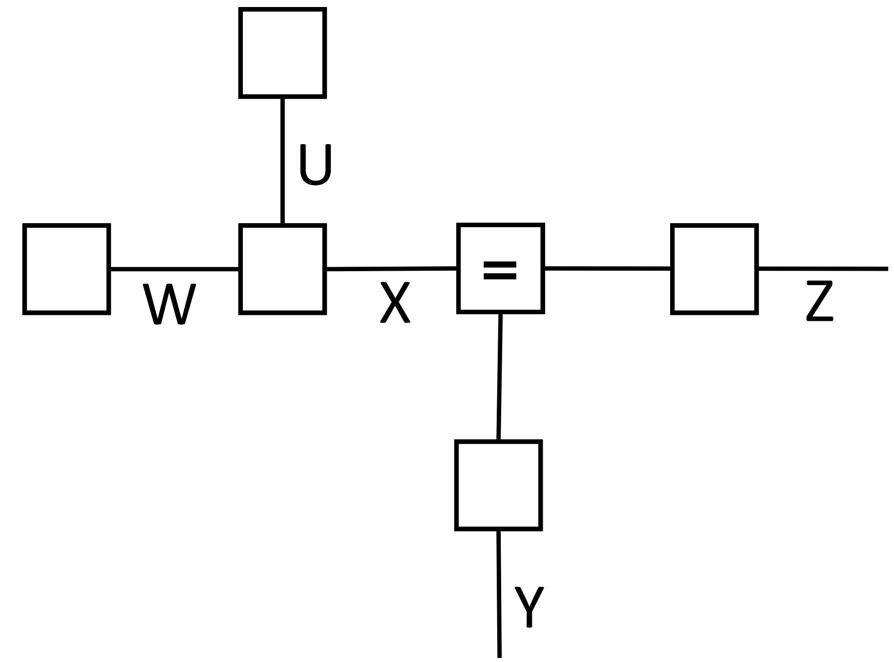
- Markov random fields
- Kalman Filters
- HMM's
- Parity Codes
- Bayesian Networks

Other notations

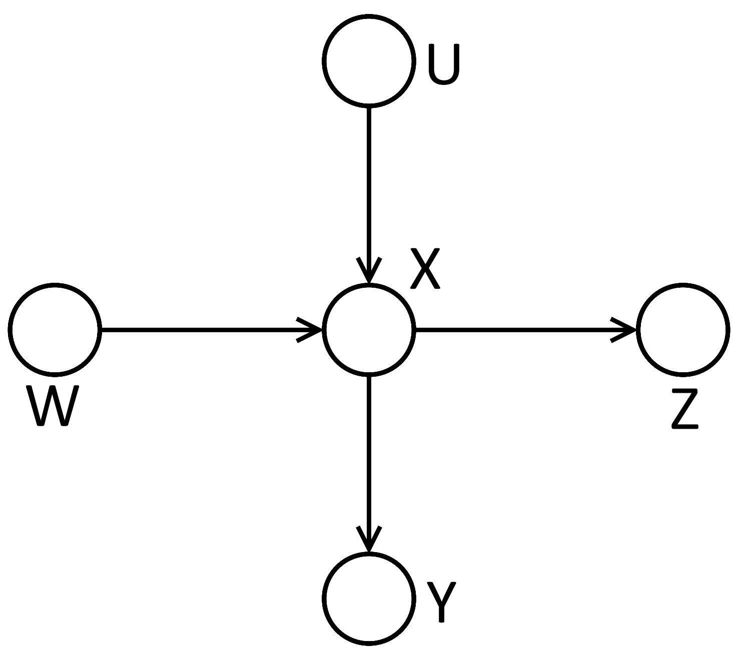
Original Factor Graph



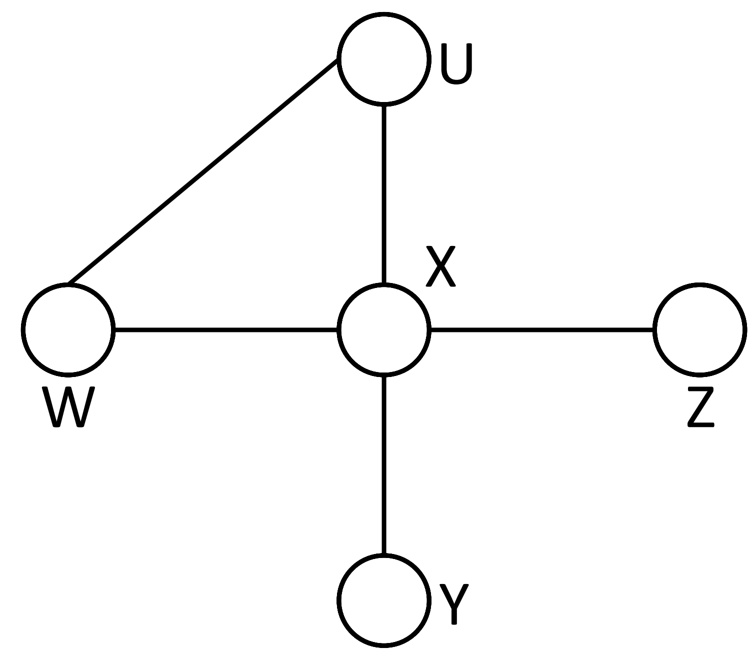
Forney-Style Factor Graph



Bayesian Network



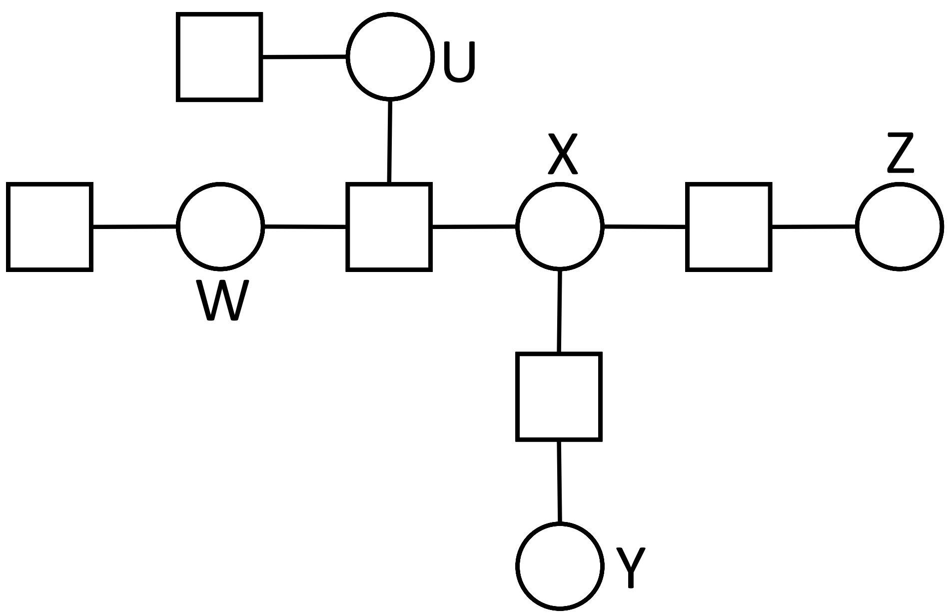
Markov Random Field



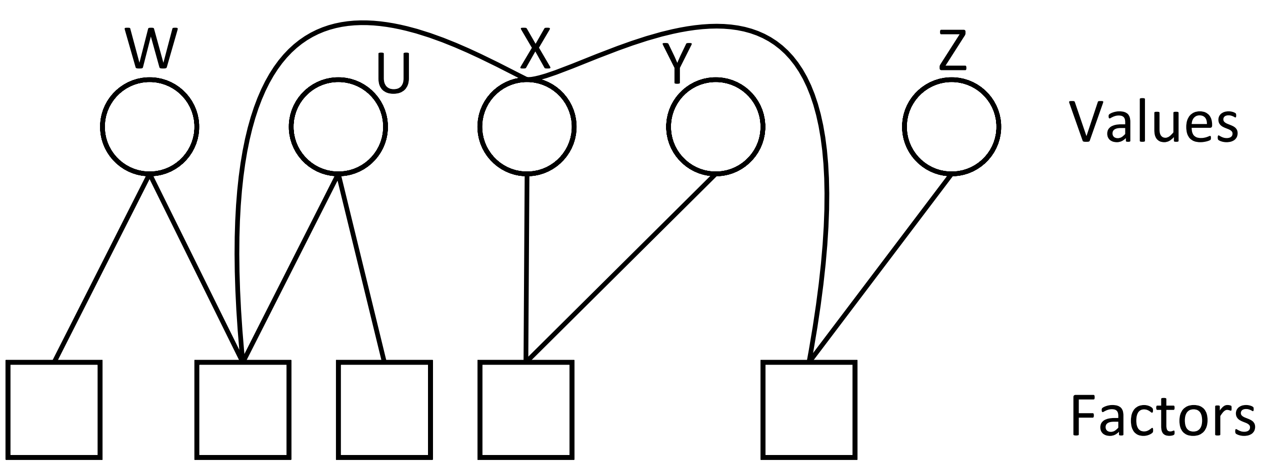
$$p(u, w, x, y, z) = p(u)p(w)p(x|u, w)p(y|x)p(z|x)$$

Bipartite Graph

Original Factor Graph



Bipartite View

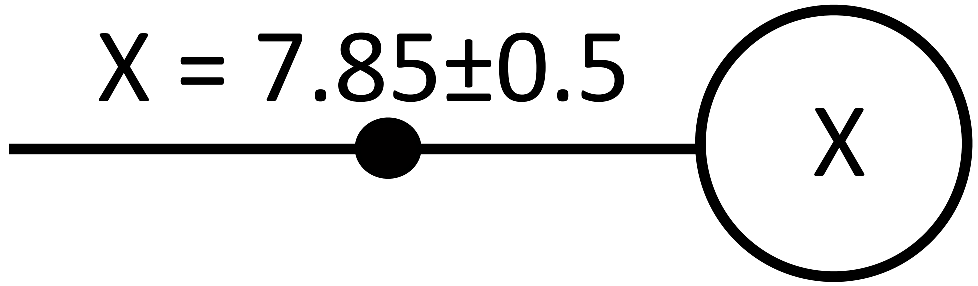


- **Factors** encapsulate constraints being imposed
- **Values** adjust to become consistent with these constraints
- Usually achieved via “**message passing**”/ gradient descent methods.

Optimization

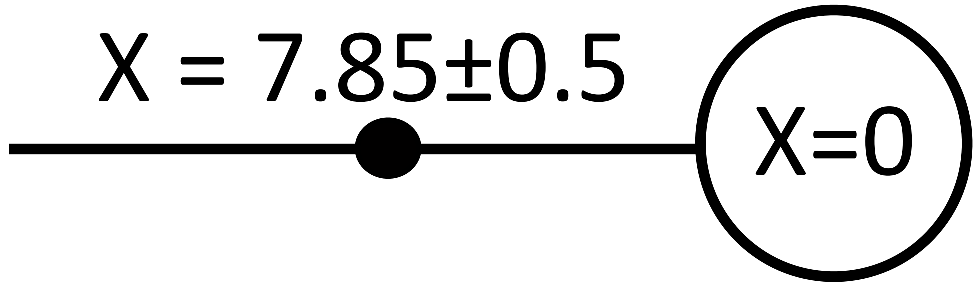
- Several algorithms are used
- Random vs “Best Guess” initialization
- Gauss-Newton Stepping
- Levenberg-Marquart
- Message Passing Algorithms
- Structure dependent/exploiting optimization strategies
  – ISAM2 (for SLAM problems)

Optimization



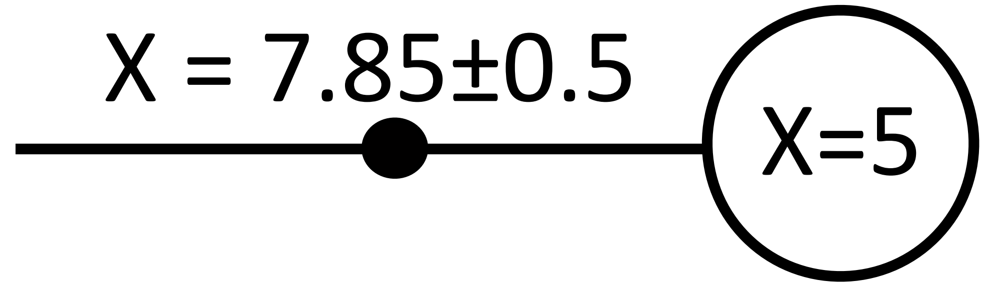
- Initialize values to a 'best guess'/random
- Iterate till convergence
  - Factors messages Values with Error
  - Values messages Factors

Optimization



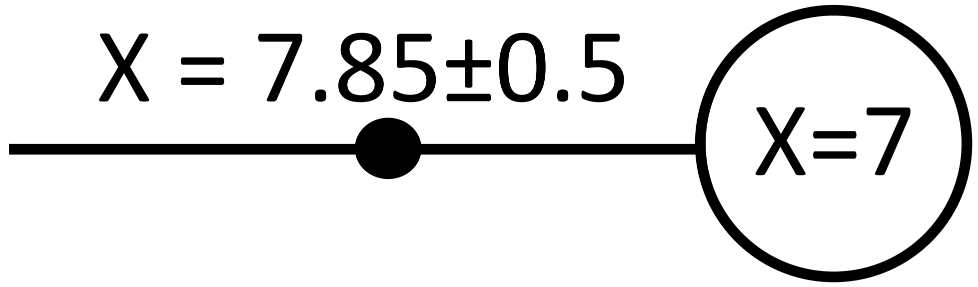
- Initialize values to a ‘best guess’/random
- Iterate till convergence
  - Factors messages Values with Error
  - Values messages Factors

Optimization



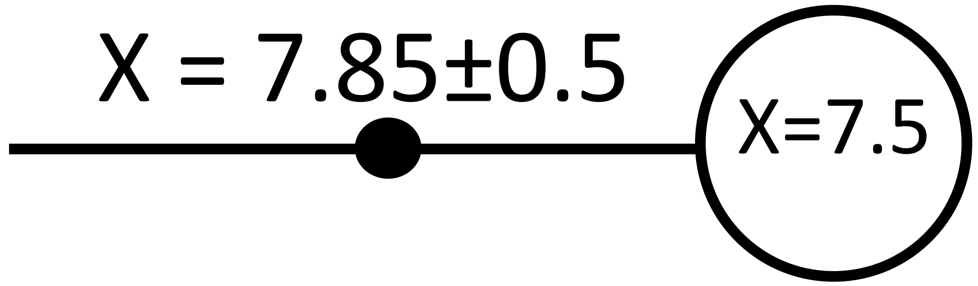
- Initialize values to a 'best guess'/random
- Iterate till convergence
  - Factors messages Values with Error
  - Values messages Factors

Optimization



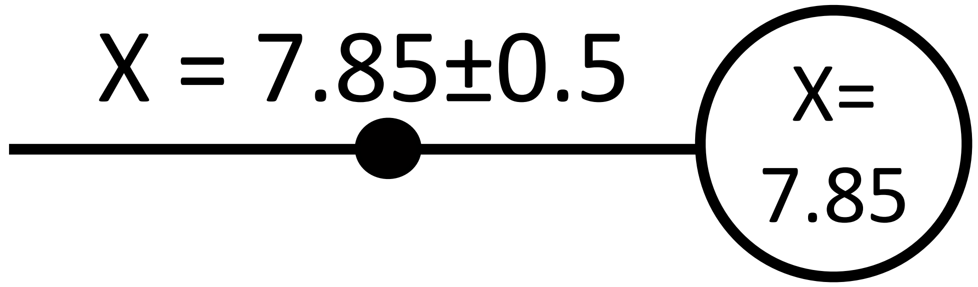
- Initialize values to a ‘best guess’/random
- Iterate till convergence
  - Factors messages Values with Error
  - Values messages Factors

Optimization



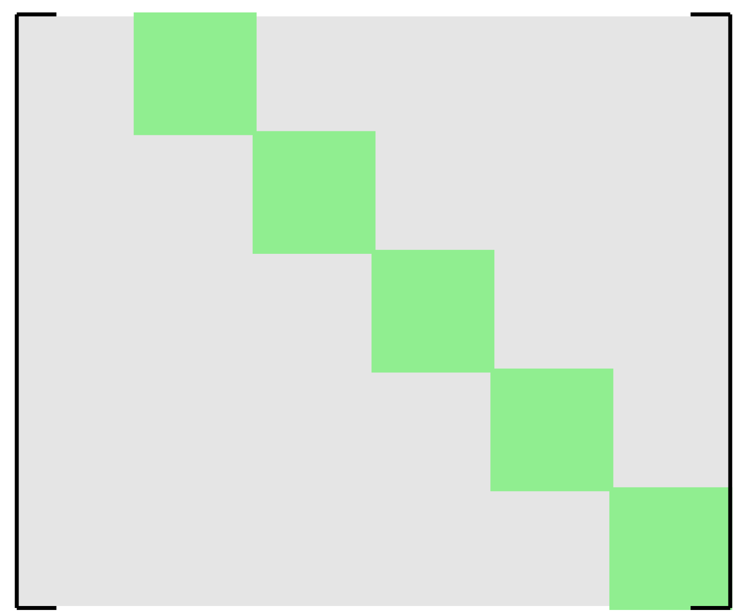
- Initialize values to a 'best guess'/random
- Iterate till convergence
  - Factors messages Values with Error
  - Values messages Factors

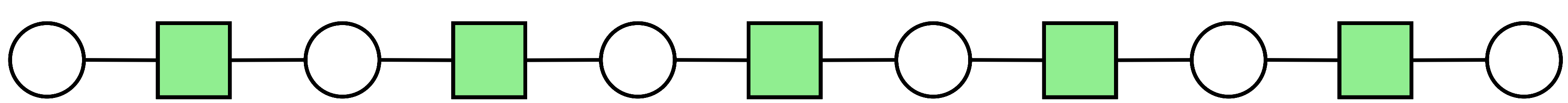
Optimization



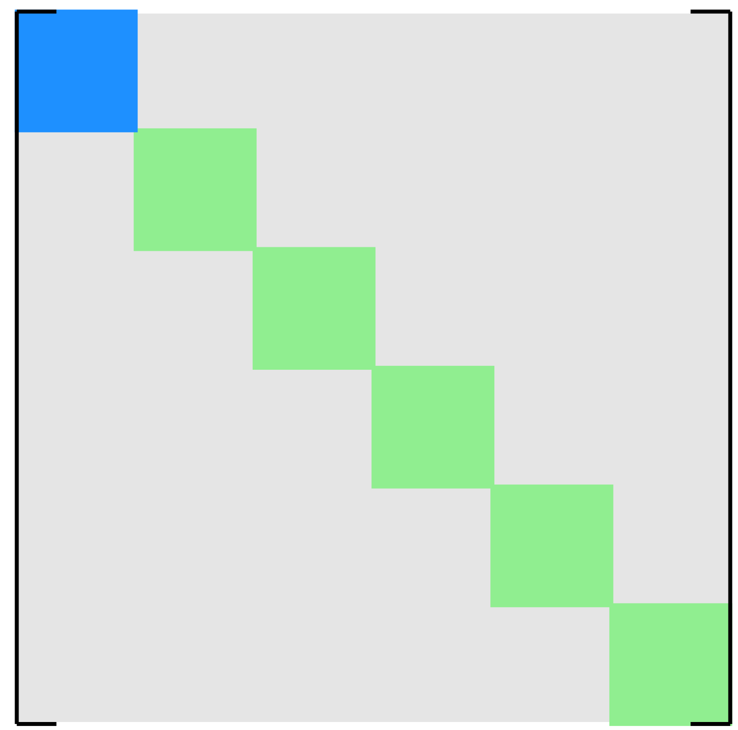
- Initialize values to a 'best guess'/random
- Iterate till convergence
  - Factors messages Values with Error
  - Values messages Factors

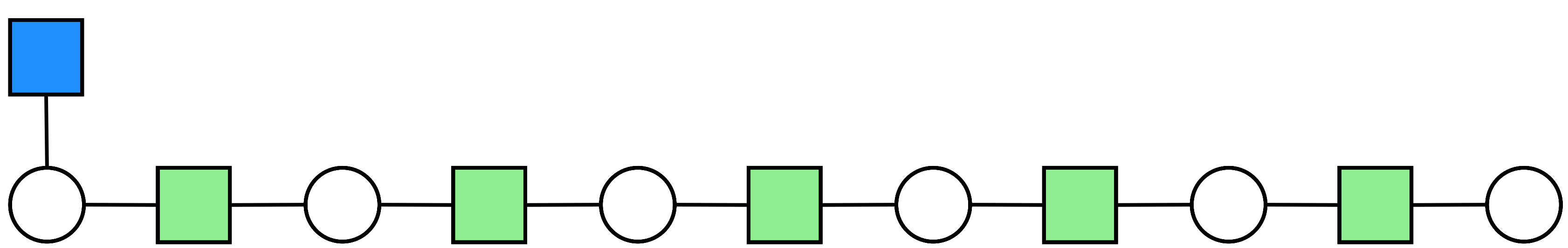
Optimization, Factorization and Sparsity

$J =$  Measurement Jacobian



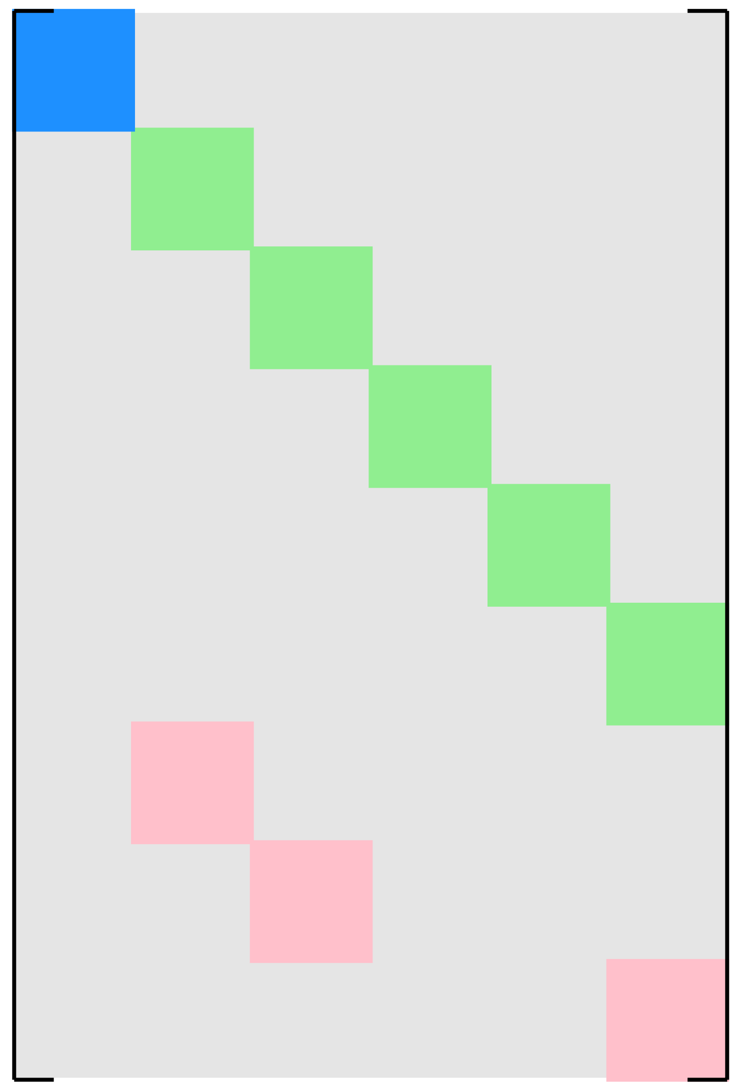
Optimization, Factorization and Sparsity

$J =$  Measurement Jacobian

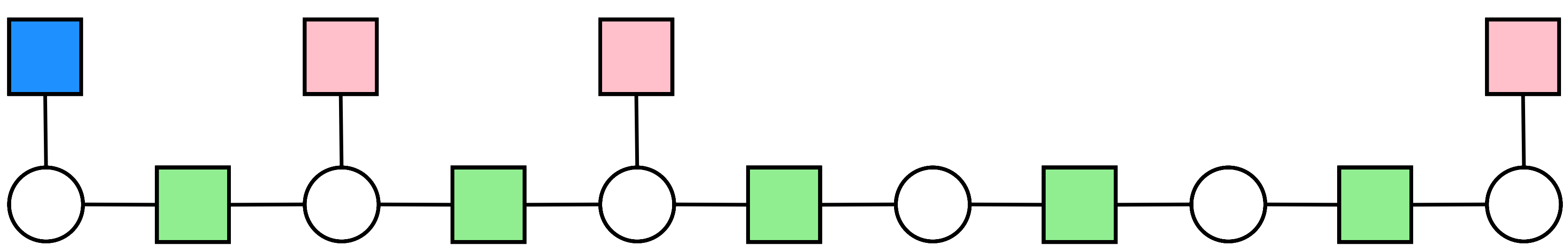


Optimization, Factorization and Sparsity

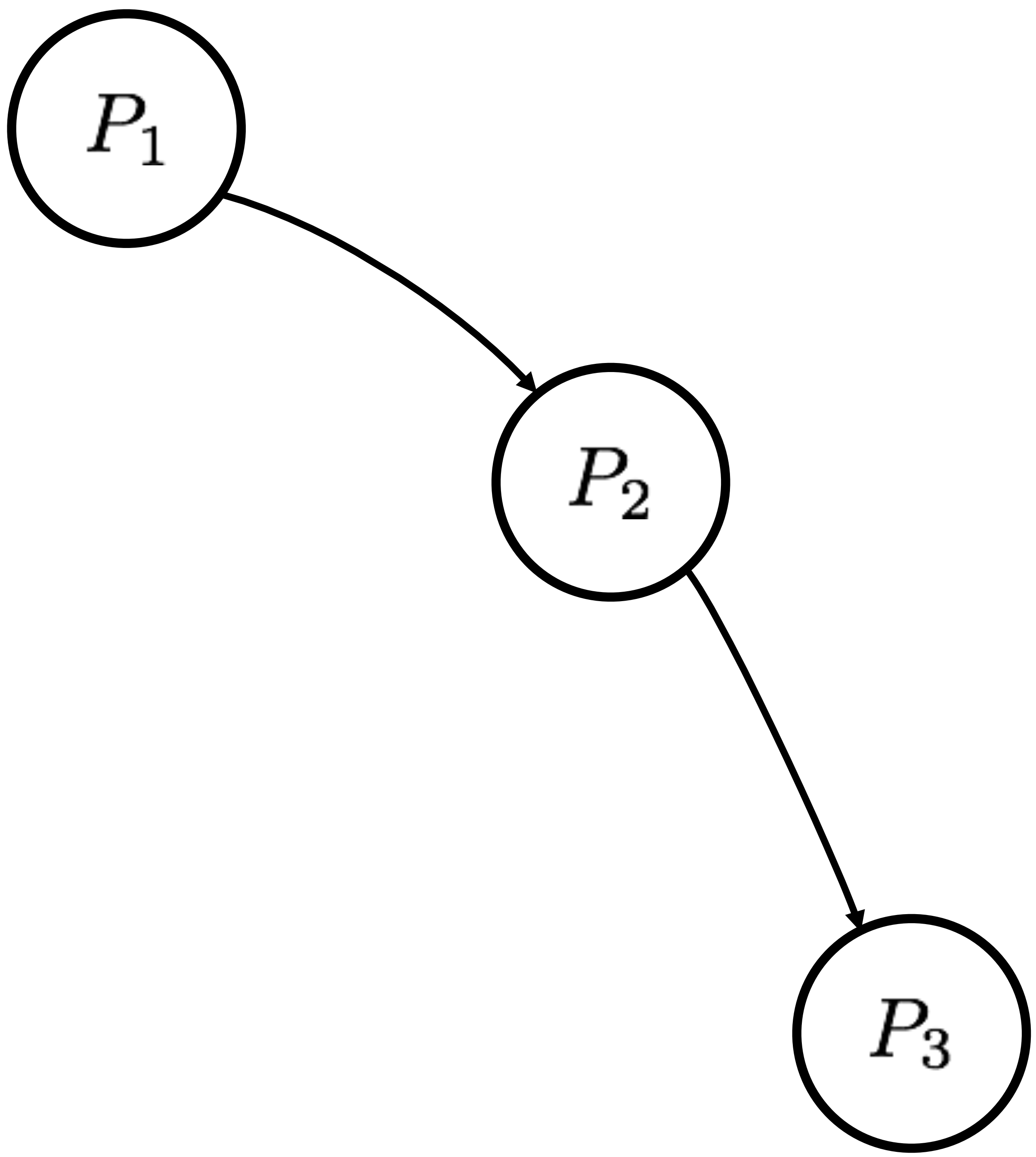
$$J =$$



Measurement Jacobian



Vehicle Pose Estimation



$$P \in \mathrm{SE}(2)\ ,\ \text{i.e.}\quad P = \begin{bmatrix} p_x \\ p_y \\ p_\theta \end{bmatrix}$$

GTSAM

- Georgia Tech Smoothing and Mapping
- State of the Art in SLAM, Bundle Adjustment, ICP

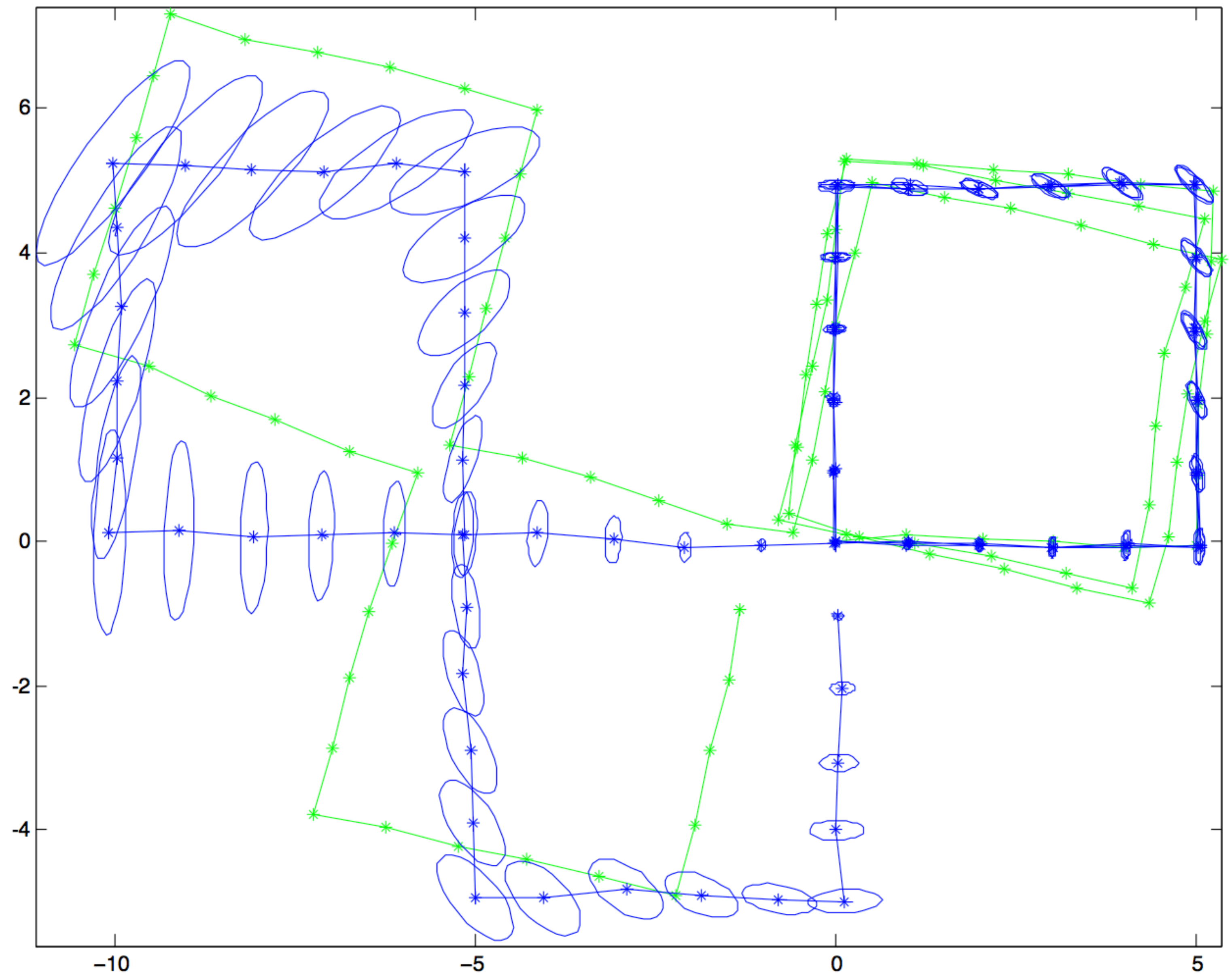


Figure 8: MATLAB plot of small Manhattan world example with 100 poses (due to Ed Olson). The initial estimate is shown in green. The optimized trajectory, with covariance ellipses, in blue.

GTSAM

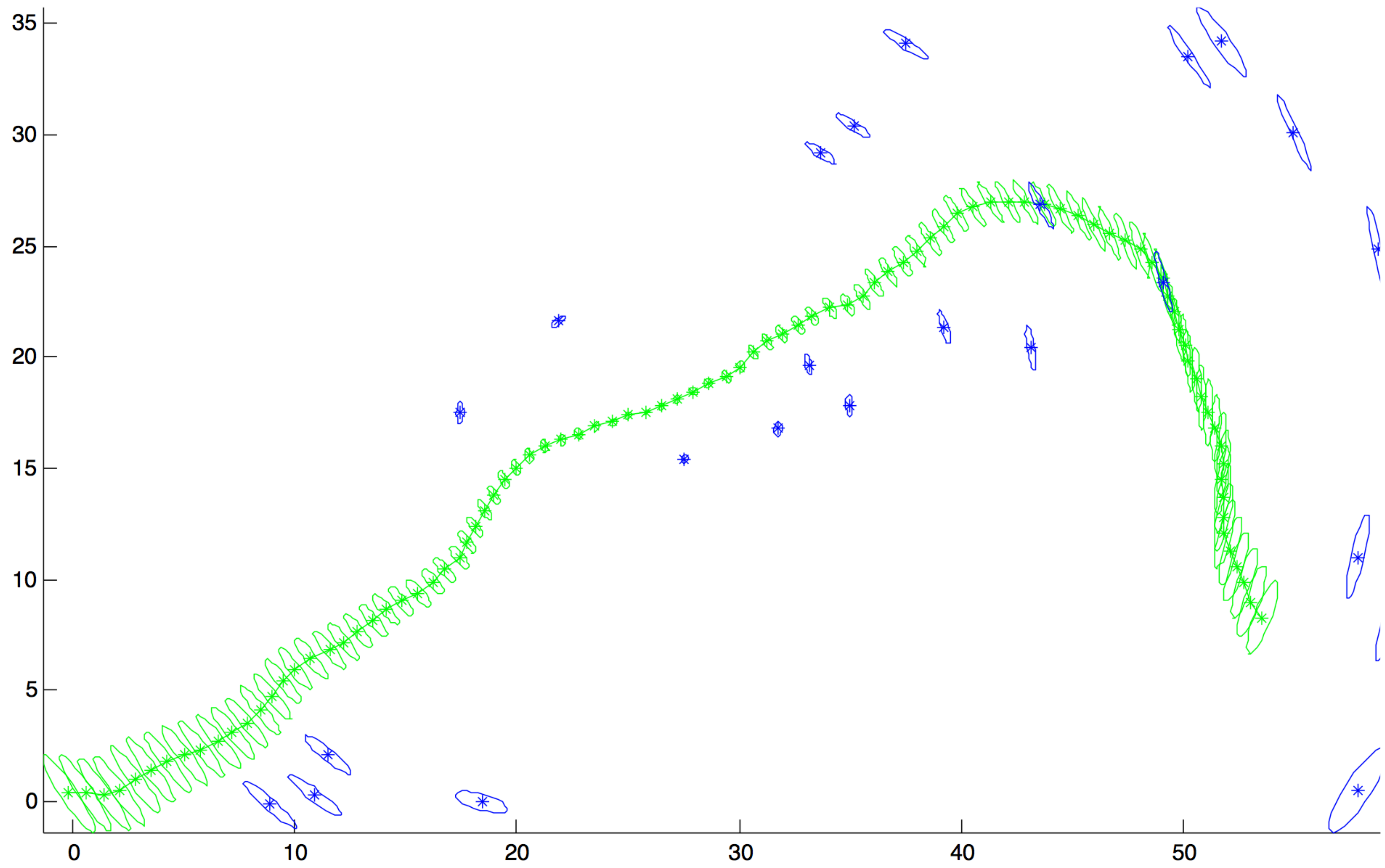


Figure 12: A larger example with about 100 poses and 30 or so landmarks, as produced by gtsam_examples/PlanarSLAMExample_graph.m

Credit: "Factor Graphs and GTSAM: A hands on introduction", Frank Dellaert

SLAM versus Localization

- No “landmarks”, rather just GPS readings
- Odometry is modeled via similar means
- Vehicle Model is far more constrained in comparison to a typical robotic agent

ARRB Vehicle Introduction

- GPS @ 1Hz
- Gyroscope, Acceleration, Odometry @ 2.0meters
  - 2-axis gyro and 2-D accel leading to somewhat under constrained problem
- Images @ 2.0-5.0 meters (unsynchronised to above)
- Random Noise including non-real-time OS errors so unrestrained arbitrary corruption of the data is possible!

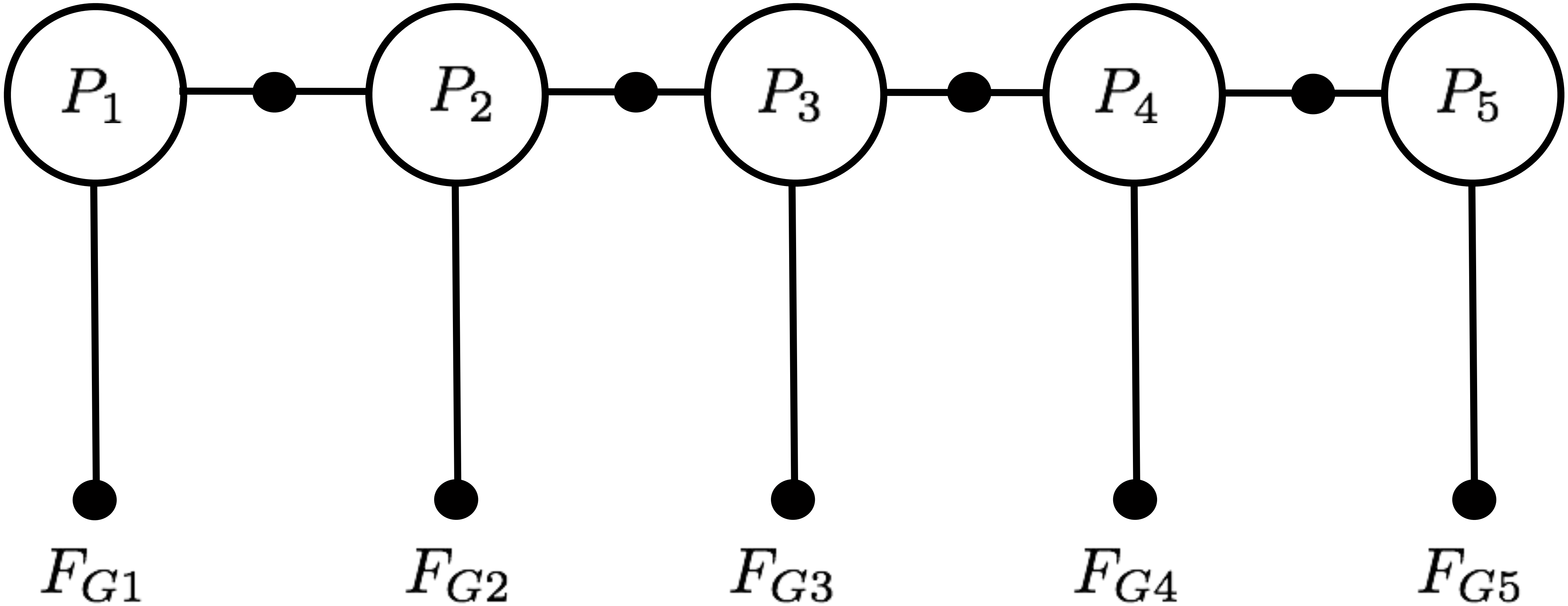
ARRB Vehicle Introduction



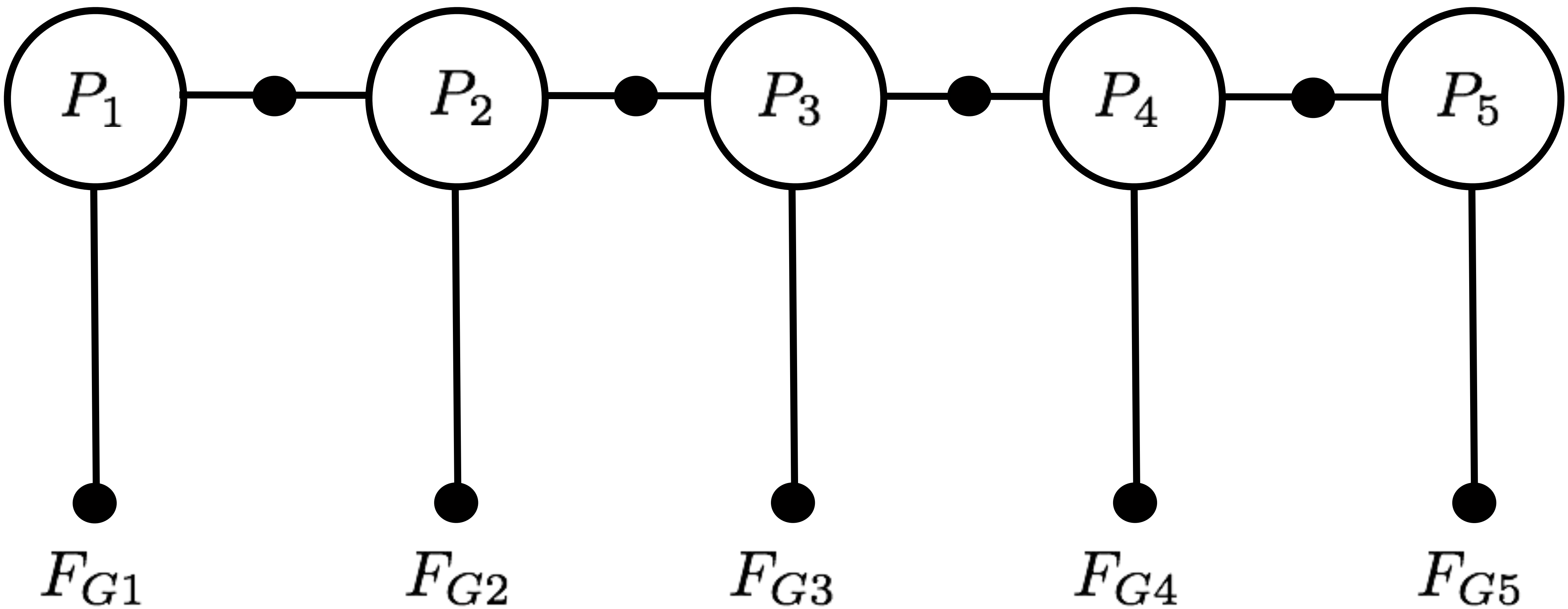

ARRB Vehicle Introduction

- GPS @ 1Hz
- Gyroscope, Acceleration, Odometry @ 2.0meters
  - 2-axis gyro and 2-D accel leading to somewhat under constrained problem
- Images @ 2.0-5.0 meters (unsynchronised to above)
- Random Noise including non-real-time OS errors so unrestrained arbitrary corruption of the data is possible!
- This is the kind of great research problem that comes about on the back of greatly incompetent design!

Pose Estimation : GPS Factor



Pose Estimation : GPS Factor



$$F_G(P;G) = exp\left\{-\frac{1}{2}||h_G(P) - G||^2_\Sigma\right\}$$

Pose Estimation : GPS Factor

$$F_G(P;G) = exp\left\{-\frac{1}{2}||h_G(P) - G||_\Sigma^2\right\}$$

$$\text{Error : } E_G(P) \triangleq h_G(P) - G$$

Pose Estimation : GPS Factor

$$F_G(P;G) = exp\left\{-\frac{1}{2}||h_G(P) - G||^2_\Sigma\right\}$$

$$\text{Error : } E_G(P) \triangleq h_G(P) - G$$

$$\text{Jacobian : } H_G = \begin{bmatrix} \frac{\delta E_{G_x}}{\delta P_x} & \frac{\delta E_{G_x}}{\delta P_y} & \frac{\delta E_{G_x}}{\delta P_\theta} \\ \frac{\delta E_{G_y}}{\delta P_x} & \frac{\delta E_{G_y}}{\delta P_y} & \frac{\delta E_{G_y}}{\delta P_\theta} \end{bmatrix}$$

Pose Estimation : GPS Factor

$$F_G(P;G) = exp\left\{-\frac{1}{2}||h_G(P) - G||^2_\Sigma\right\}$$

$$\text{Error}: \; E_G(P) \triangleq h_G(P) - G$$

$$\text{Jacobian}: \; H_G = \begin{bmatrix} \frac{\delta E_{G_x}}{\delta P_x} & \frac{\delta E_{G_x}}{\delta P_y} & \boxed{\begin{matrix} \frac{\delta E_{G_x}}{\delta P_\theta} \\ \frac{\delta E_{G_y}}{\delta P_\theta} \end{matrix}} \\ \frac{\delta E_{G_y}}{\delta P_x} & \frac{\delta E_{G_y}}{\delta P_y} & \end{bmatrix}$$

Pose Estimation : GPS Factor

$$F_G(P;G) = exp\left\{-\frac{1}{2}||h_G(P) - G||^2_\Sigma\right\}$$

$$\text{Error}: \ E_G(P) \triangleq h_G(P) - G$$

$$\text{Jacobian}: \ H_G = \begin{bmatrix} \frac{\delta E_{G_x}}{\delta P_x} & \frac{\delta E_{G_x}}{\delta P_y} & 0 \\ \frac{\delta E_{G_y}}{\delta P_x} & \frac{\delta E_{G_y}}{\delta P_y} & 0 \end{bmatrix}$$

Pose Estimation : GPS Factor

$$F_G(P;G) = exp\left\{-\frac{1}{2}||h_G(P) - G||^2_\Sigma\right\}$$

$$\text{Error : } E_G(P) \triangleq h_G(P) - G$$

$$\text{Jacobian : } H_G = \begin{bmatrix} \frac{\delta E_{G_x}}{\delta P_x} & \boxed{\frac{\delta E_{G_x}}{\delta P_y}} & 0 \\ \boxed{\frac{\delta E_{G_y}}{\delta P_x}} & \frac{\delta E_{G_y}}{\delta P_y} & 0 \end{bmatrix}$$

Pose Estimation : GPS Factor

$$F_G(P;G) = exp\left\{-\frac{1}{2}||h_G(P) - G||_\Sigma^2\right\}$$

$$\text{Error : } E_G(P) \triangleq h_G(P) - G$$

$$\text{Jacobian : } H_G = \begin{bmatrix} \frac{\delta E_{G_x}}{\delta P_x} & 0 & 0 \\ 0 & \frac{\delta E_{G_y}}{\delta P_y} & 0 \end{bmatrix}$$

Pose Estimation : GPS Factor

$$F_G(P;G) = exp\left\{-\frac{1}{2}||h_G(P) - G||^2_\Sigma\right\}$$

$$\text{Error : } E_G(P) \triangleq h_G(P) - G$$

$$\text{Jacobian : } H_G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Pose Estimation : GPS Factor

$$F_G(P;G) = exp\left\{-\frac{1}{2}||h_G(P) - G||^2_\Sigma\right\}$$

$$\text{Error}: \; E_G(P) \triangleq h_G(P) - G$$

$$\text{Jacobian}: \; H_G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$Covariance : \Sigma_{E_G} = \begin{bmatrix} \sigma_x \\ \sigma_y \end{bmatrix} = \begin{bmatrix} 10m \\ 10m \end{bmatrix}$$

Pose Estimation : GPS Factor

$$F_G(P;G) = exp\left\{-\frac{1}{2}||h_G(P) - G||^2_\Sigma\right\}$$

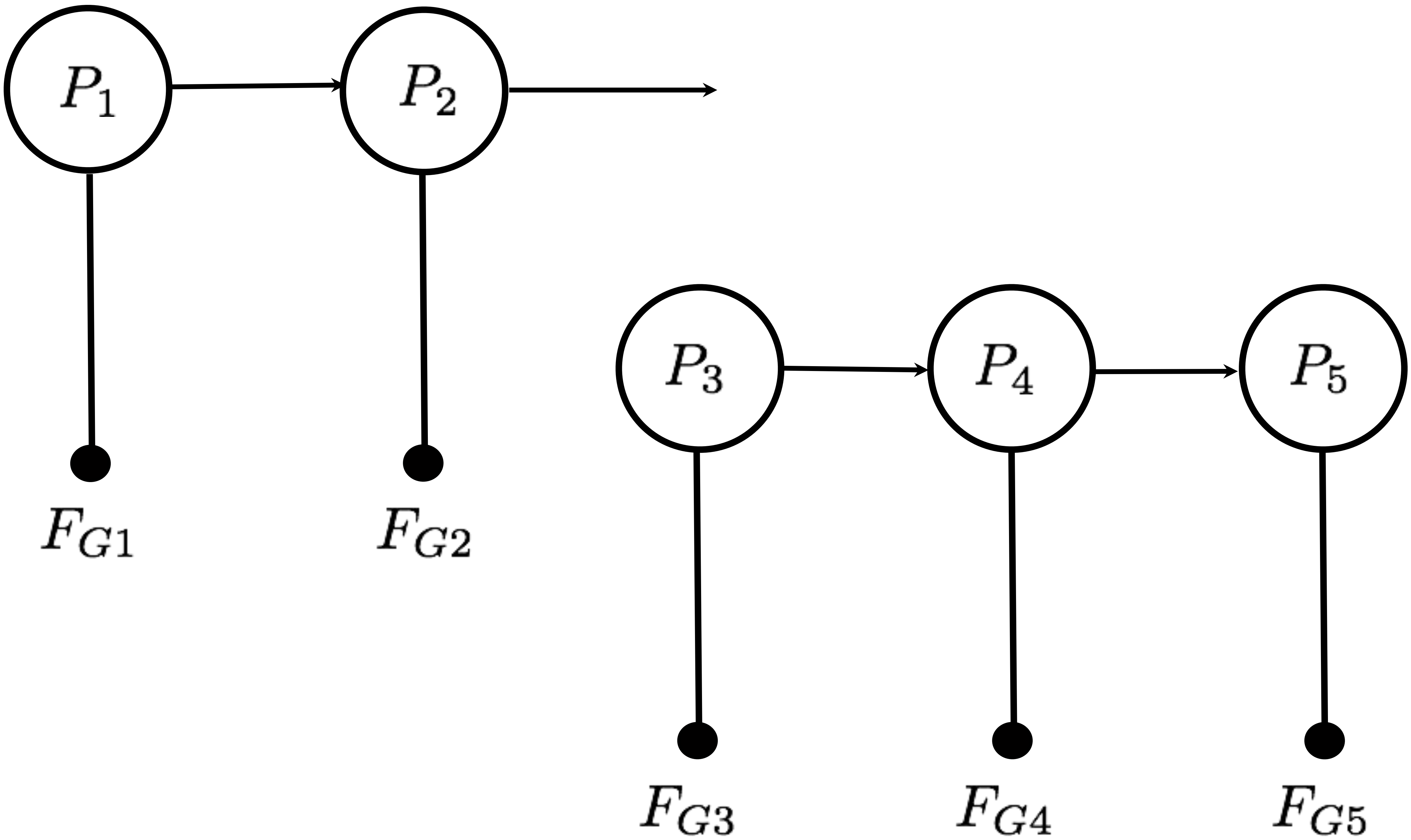
$$\text{Error}\ :\ E_G(P) \triangleq h_G(P) - G$$

$$\text{Jacobian}\ :\ H_G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

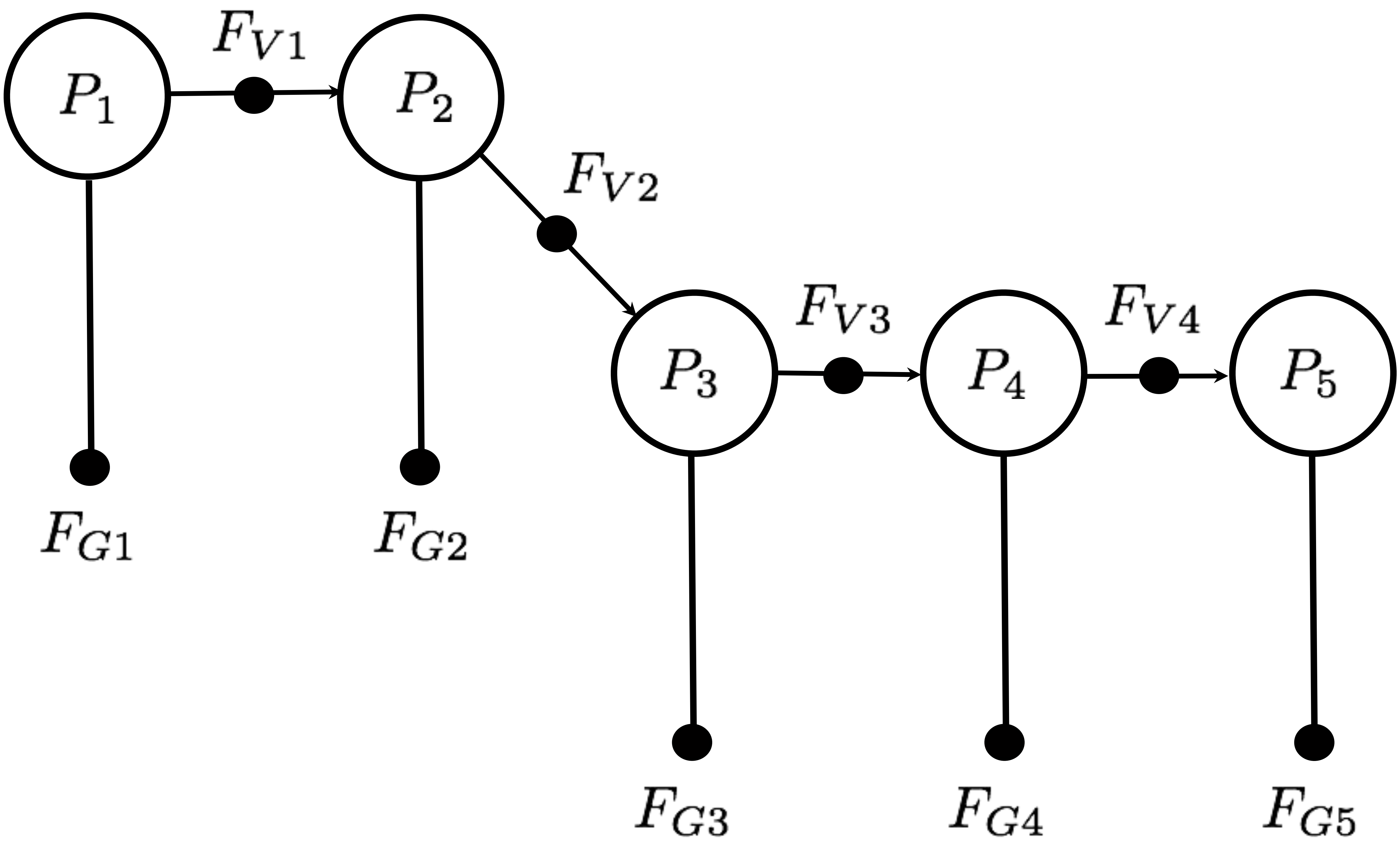
$$Covariance : \Sigma_{E_G} = \begin{bmatrix} \sigma_x \\ \sigma_y \end{bmatrix} = \begin{bmatrix} 10m \\ 10m \end{bmatrix}$$

Once you have defined all these you are ready to code your Factor in GTSAM

Pose Estimation : GPS Discontinuity



Pose Estimation : Vehicle Model Factor



Pose Estimation : Vehicle Model Factor

Prediction with binary factor requires more than just pose information.
Therefore we augment P with additional variables to track; time, speed and steering angle.

$$P = \begin{bmatrix} x \\ y \\ \theta \\ t \\ s \\ \omega \end{bmatrix}$$

Pose Estimation : Vehicle Model Factor

Prediction with binary factor requires more than just pose information.
Therefore we augment P with additional variables to track; time, speed and steering angle.

$$P = \begin{bmatrix} x \\ y \\ \theta \\ t \\ s \\ \omega \end{bmatrix}$$

$$h_x^*(P_1, P_2) = \frac{1}{\omega_1} \sin\left(\omega_1 s_1 (t_2 - t_1)\right)$$

$$h_y^*(P_1, P_2) = \frac{1}{\omega_1} \left(1 - \cos\left(\omega_1 s_1 (t_2 - t_1)\right)\right)$$

$$h_\theta^*(P_1, P_2) = \omega_1 s_1 (t_2 - t_1)$$

$$h_s(P_1, P_2) = s_2 - s_1$$

$$h_\omega(P_1, P_2) = \omega_2 - \omega_1$$

Pose Estimation : Vehicle Model Factor

Prediction with binary factor requires more than just pose information.
Therefore we augment P with additional variables to track; time, speed and steering angle.

$$P = \begin{bmatrix} x \\ y \\ \theta \\ t \\ s \\ \omega \end{bmatrix}$$

$$\boxed{\begin{aligned} h_x^*(P_1, P_2) &= \frac{1}{\omega_1} \sin\left(\omega_1 s_1 (t_2 - t_1)\right) \\ h_y^*(P_1, P_2) &= \frac{1}{\omega_1} \left(1 - \cos\left(\omega_1 s_1 (t_2 - t_1)\right)\right) \\ h_\theta^*(P_1, P_2) &= \omega_1 s_1 (t_2 - t_1) \end{aligned}}$$

Pose Prediction Model

$$h_s(P_1, P_2) = s_2 - s_1$$

$$h_\omega(P_1, P_2) = \omega_2 - \omega_1$$

Pose Estimation : Vehicle Model Factor

Prediction with binary factor requires more than just pose information.
Therefore we augment P with additional variables to track; time, speed and steering angle.

$$P = \begin{bmatrix} x \\ y \\ \theta \\ t \\ s \\ \omega \end{bmatrix}$$

$$h^*_x(P_1, P_2) = \frac{1}{\omega_1} \sin\left(\omega_1 s_1 (t_2 - t_1)\right)$$

???

$$h^*_y(P_1, P_2) = \frac{1}{\omega_1} \left(1 - \cos\left(\omega_1 s_1 (t_2 - t_1)\right)\right)$$

$$h^*_\theta(P_1, P_2) = \omega_1 s_1 (t_2 - t_1)$$

Pose Prediction Model

$$h_s(P_1, P_2) = s_2 - s_1$$

$$h_\omega(P_1, P_2) = \omega_2 - \omega_1$$

Pose Estimation : Vehicle Model Factor

Prediction with binary factor requires more than just pose information.
Therefore we augment P with additional variables to track; time, speed and steering angle.

$$P = \begin{bmatrix} x \\ y \\ \theta \\ t \\ s \\ \omega \end{bmatrix}$$

$$h_x^*(P_1, P_2) = \frac{1}{\omega_1} \sin\left(\omega_1 s_1 (t_2 - t_1)\right)$$

$$h_y^*(P_1, P_2) = \frac{1}{\omega_1} \Big(1 - \cos\left(\omega_1 s_1 (t_2 - t_1)\right)\Big)$$

$$h_\theta^*(P_1, P_2) = \omega_1 s_1 (t_2 - t_1)$$

Damping

$$\boxed{\begin{aligned} h_s(P_1, P_2) &= s_2 - s_1 \\ h_\omega(P_1, P_2) &= \omega_2 - \omega_1 \end{aligned}}$$

Pose Estimation : Vehicle Model Factor

Prediction with binary factor requires more than just pose information.
Therefore we augment P with additional variables to track; time, speed and steering angle.

$$P = \begin{bmatrix} x \\ y \\ \theta \\ t \\ s \\ \omega \end{bmatrix}$$

$$h_x^*(P_1, P_2)$$

$$h_y^*(P_1, P_2)$$

$$h_\theta^*(P_1, P_2)$$

$$h_s(P_1, P_2)$$

$$h_\omega(P_1, P_2)$$

And compute the 5×6 Jacobians for P_1 & P_2!

Pose Estimation : Vehicle Model Factor

Prediction with binary factor requires more than just pose information.
Therefore we augment P with additional variables to track time, speed and steering angle.

$$P = \begin{bmatrix} x \\ y \\ \theta \\ t \\ s \\ \omega \end{bmatrix}$$

$$h_x^*(P_1, P_2)$$

$$h_y^*(P_1, P_2)$$

$$h_\theta^*(P_1, P_2)$$

$$h_s(P_1, P_2)$$

$$h_\omega(P_1, P_2)$$

And compute the 5×6 Jacobians for P_1 & P_2!

$$H_{P_i} = \begin{bmatrix} \frac{\delta h_x^*}{\delta x_i} & \frac{\delta h_x^*}{\delta y_i} & \frac{\delta h_x^*}{\delta \theta_i} & \frac{\delta h_x^*}{\delta t_i} & \frac{\delta h_x^*}{\delta s_i} & \frac{\delta h_x^*}{\delta w_i} \\ \frac{\delta h_y^*}{\delta x_i} & \ddots & & & & \vdots \\ \frac{\delta h_\theta^*}{\delta x_i} & & & \ddots & & \\ \frac{\delta h_s}{\delta x_i} & & & & \ddots & \\ \frac{\delta h_\omega}{\delta x_i} & \cdots & & & \ddots & \frac{\delta h_\omega}{\delta w_i} \end{bmatrix}$$

in code!

Pose Estimation : Vehicle Model Factor

Prediction with binary factor requires more than just pose information.
Therefore we augment P with additional variables to track; time, speed and steering angle.

$$P = \begin{bmatrix} x \\ y \\ \theta \\ t \\ s \\ \omega \end{bmatrix}$$

$$h_x^*(P_1, P_2)$$

$$h_y^*(P_1, P_2)$$

$$h_\theta^*(P_1, P_2)$$

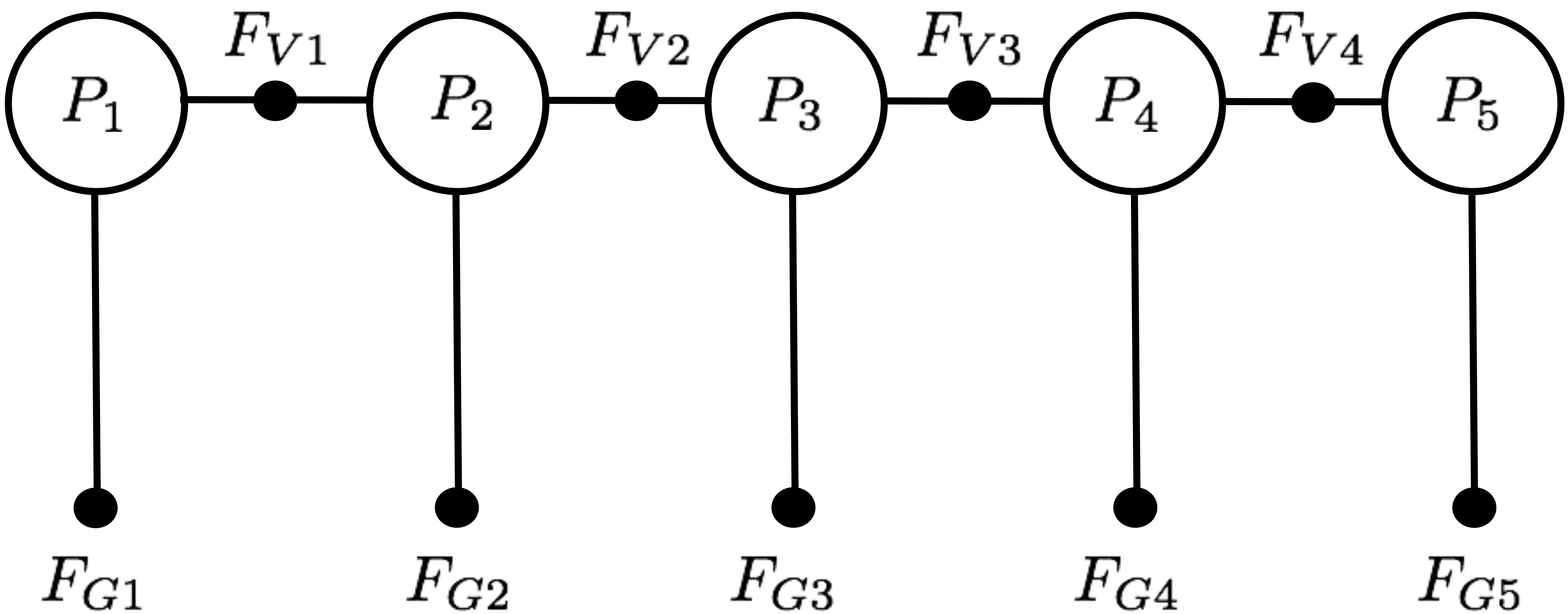
$$h_s(P_1, P_2)$$

$$h_\omega(P_1, P_2)$$

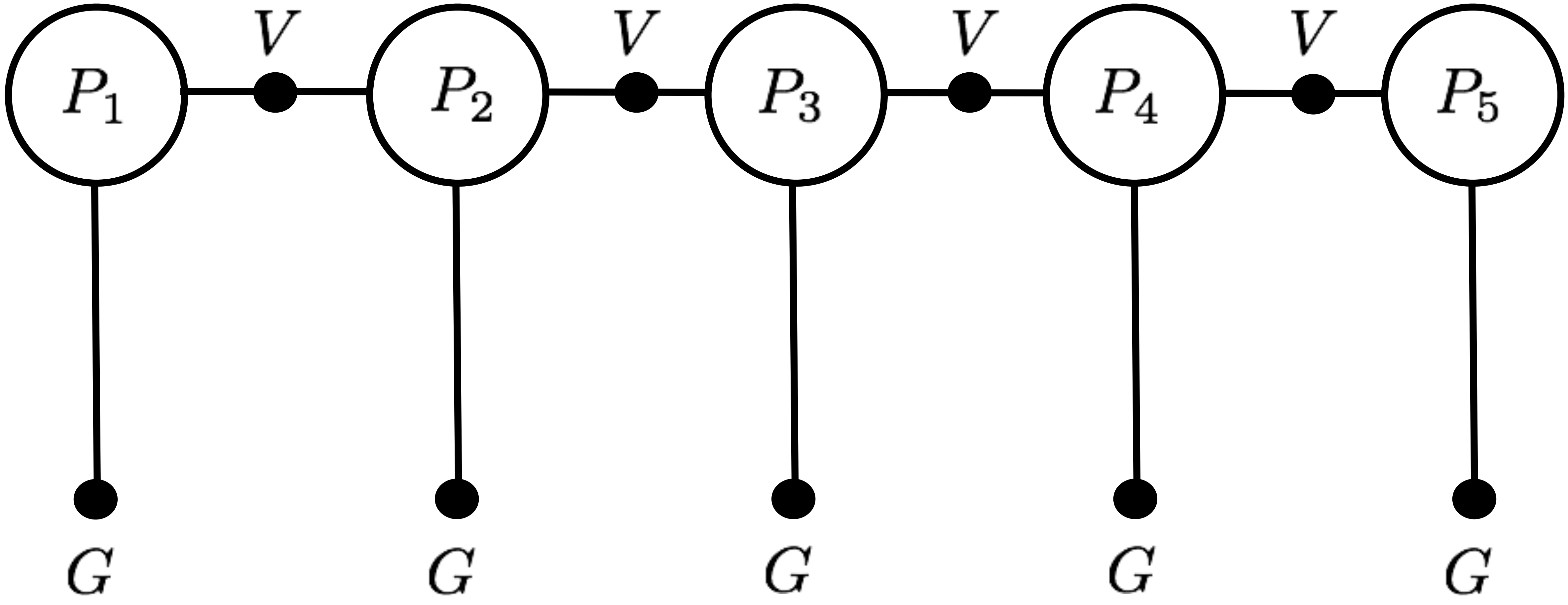
And compute the 5×6 Jacobians for P_1 & P_2!

or since GTSAM 2.3 use numerical derivatives!

Pose Estimation : GPS+Vehicle...

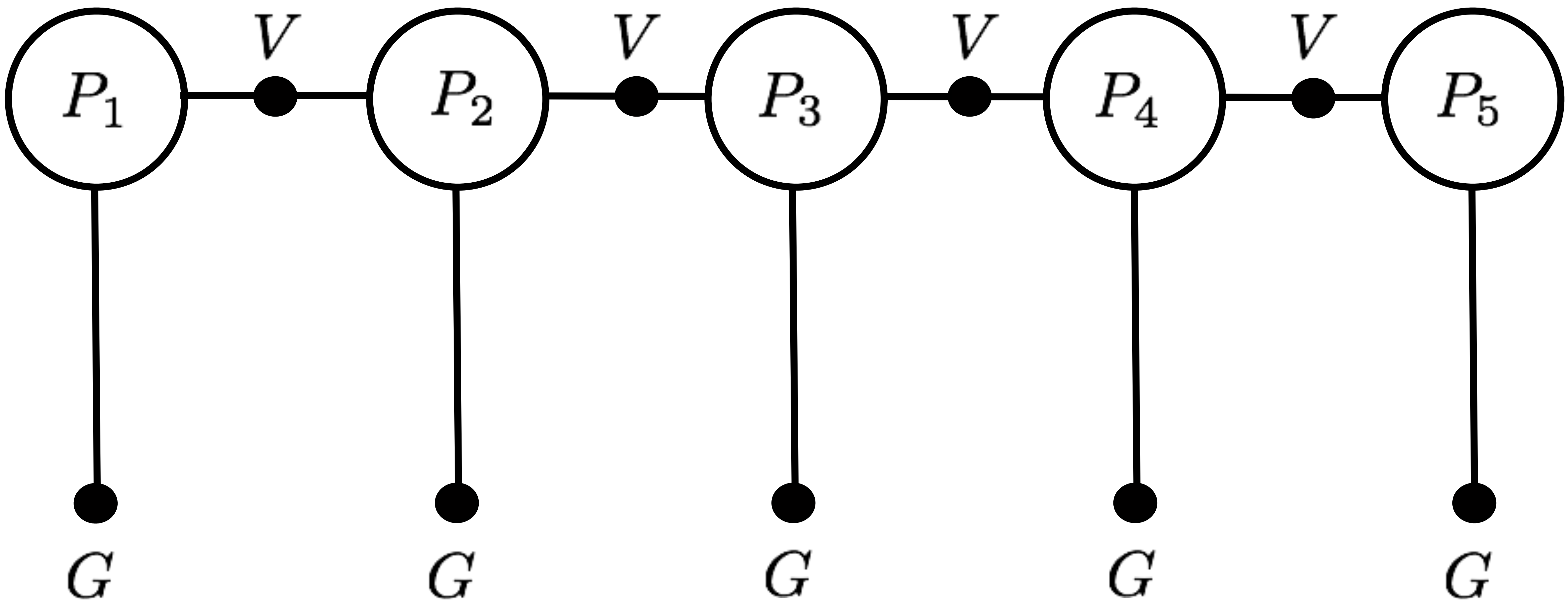


Pose Estimation : GPS+Vehicle...

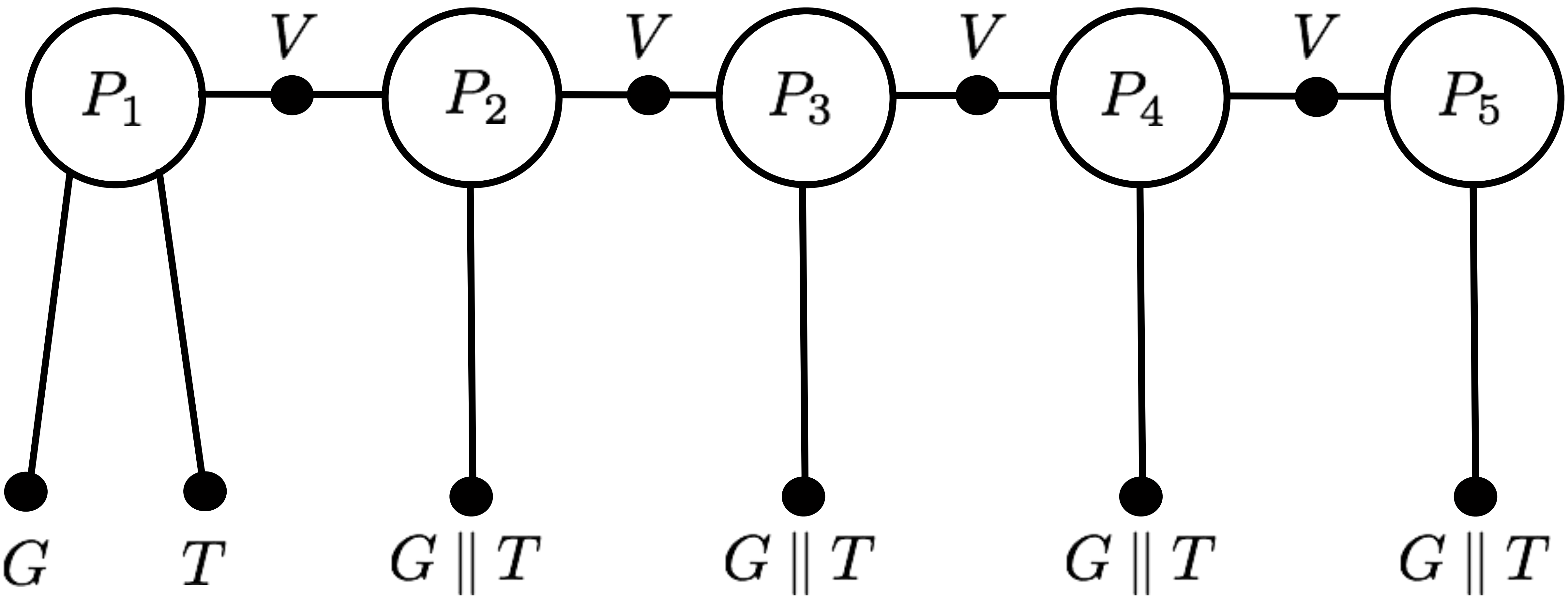


Pose Estimation : GPS+Vehicle...

Indeterminate System without time being constrained.

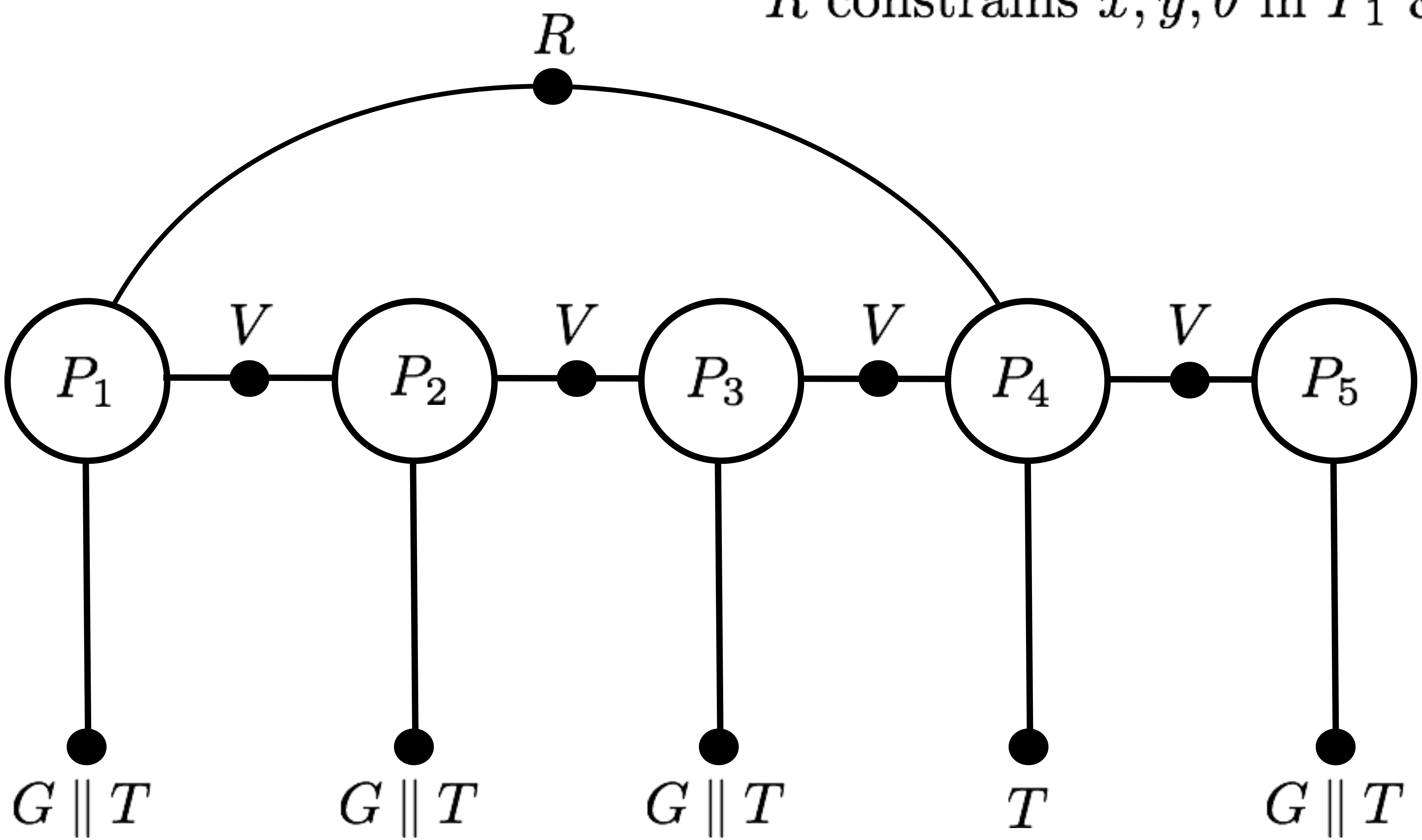


Pose Estimation : GPS+Vehicle+Time

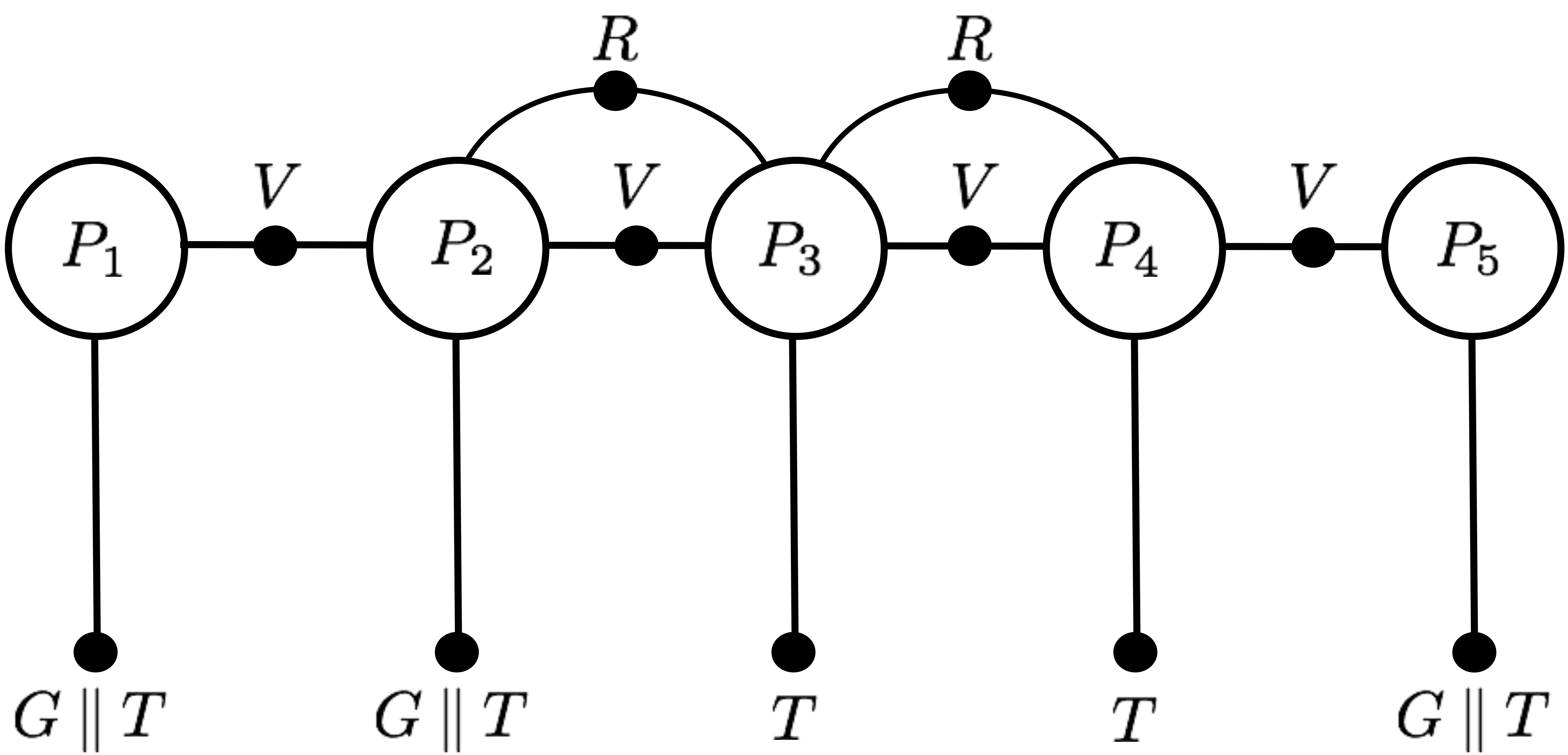


Pose Estimation : GyroOdo Factor (R)

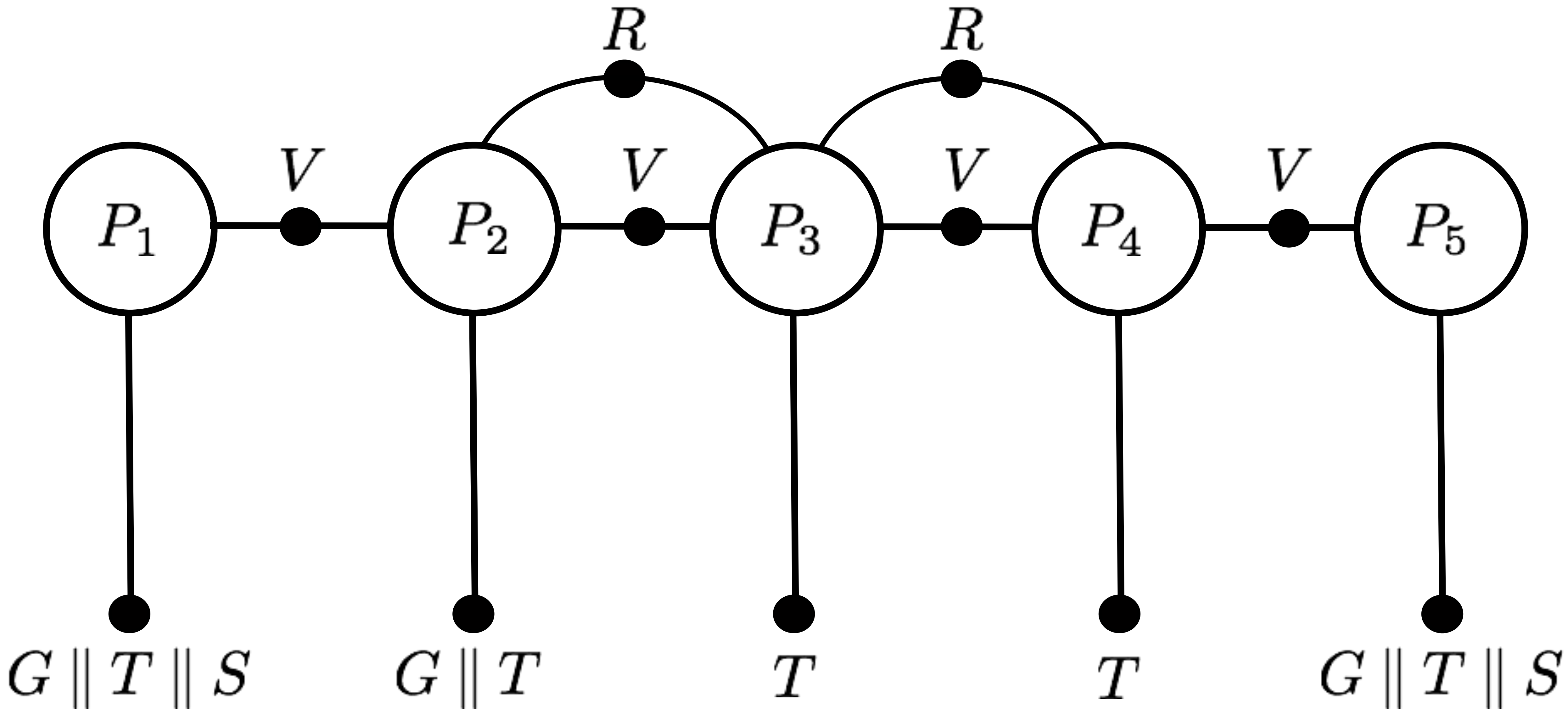
R constrains x, y, θ in P_1 & P_4



Pose Estimation : GyroOdo Factor (R)



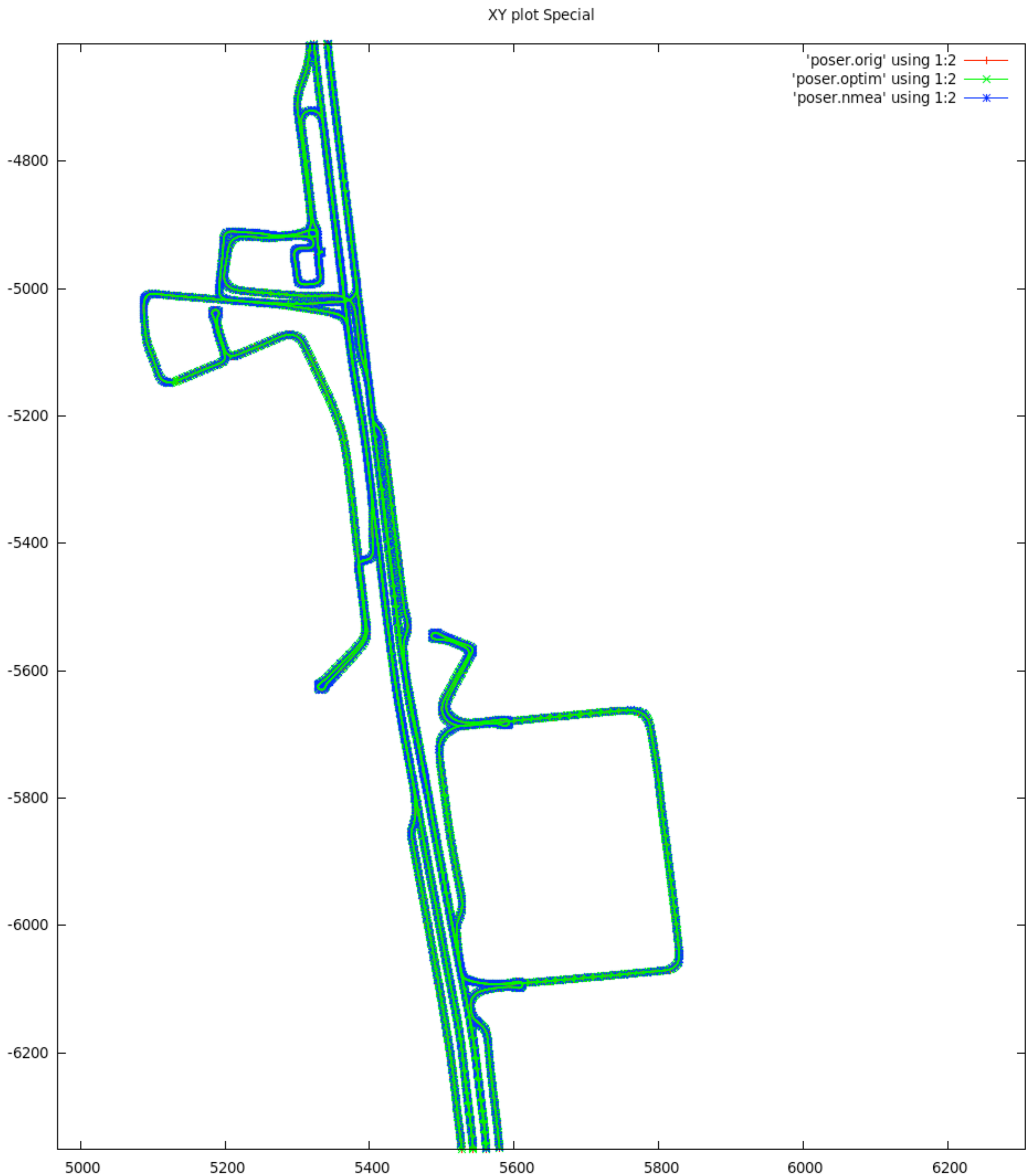
Pose Estimation : Stationary Factor (S)



Optimization

- Levenberg-Marquardt, iSAM, iSAM2, others
- Initialization sensitivities can require somewhat complex ‘rules of thumb’ to avoid undesired minima
- Speed can be an issue
- Without ground truth data or manual checking its always possible there are unpleasant surprises in the optimized output

Results : “Poser”



Results : “Poser” - more than 2 mile dropout

Results : “Poser” - tested on > 40,000miles

Results : “Poser” - tested on > 40,000miles

2 known fail cases :(

GICP : Kinect Sensor

https://www.youtube.com/watch?v=TY99Y_I_egg

Factor Graphs : Round Up

- Factor Graphs are great for expressing formally defined relationships (e.g. in SLAM we 'know' how GPS positions should relate to real world locations)
- Highly General Tool – but its useful for building specific models/capturing prior knowledge you can formalize
- Contrast with Neural Networks – both are graphical models but Neural Networks work to discover the unknown (or you are lazy) relationships as opposed to known ones.

Resources

- GTSAM - https://collab.cc.gatech.edu/borg/download
- "Factor Graphs and GTSAM: A hands on introduction", Frank Dellaert
- Michael Kaess – GTSAM contributor now at CMU-RI (see https://www.ri.cmu.edu/video_view.html?video_id=129&menu_id=387)
- G2O – an alternative library for similar factor graph optimizations
- Dimple – Graphical Modelling Tool http://dimple.probprog.org/