Ensemble-based decision support system for geosteering

NFES Stavanger Monthly Technical Meeting — May 2018

Sergey Alyaev[1],
Reidar Bratvold[2],
Erich Suter, Geir Evensen, Xiaodong Luo and Erlend Vefring [1]

[1]IRIS

[2]University of Stavanger

May 7, 2018

Geosteering

What is geosteering?

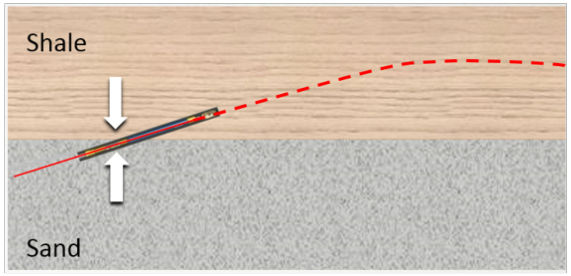
Geosteering is **the optimal placement (1)** of a wellbore based on the results of **realtime downhole geological and geophysical logging measurements (2)** rather than three-dimensional targets in space.

[Wikipedia]

History of geosteering: realtime EM measurements (2)

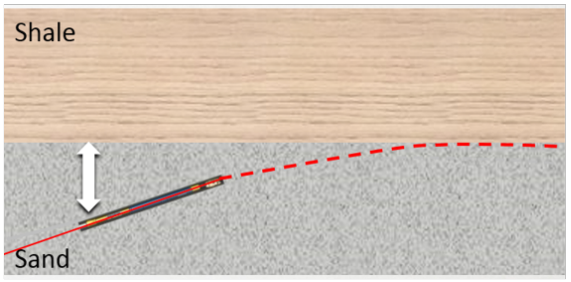
First run 1996

Simple correlation work



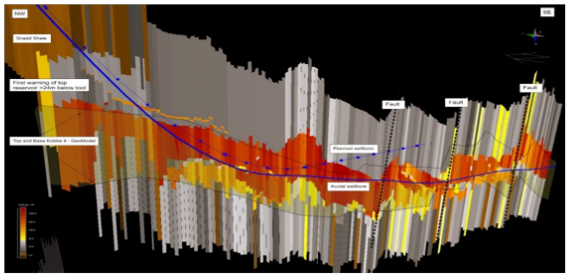
Deep Azimuthal Resistivity 2006

Proper Pro-active geosteering commences



Extra Deep Azimuthal Res. 2012

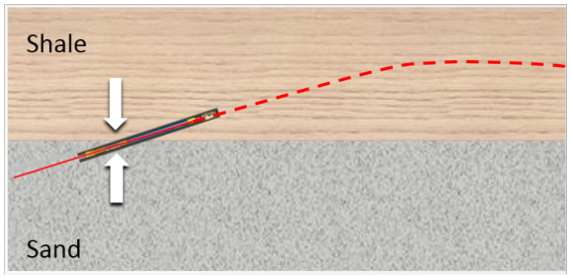
Opens the doors for proper mapping and analyisis



History of geosteering: realtime EM measurements (2)

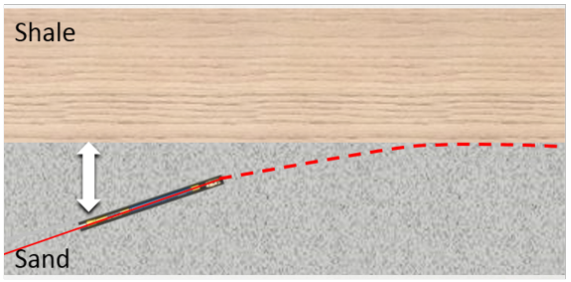
First run 1996

Simple correlation work



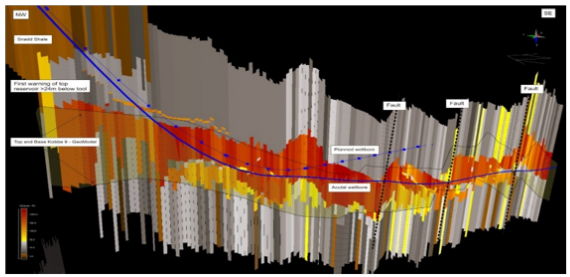
Deep Azimuthal Resistivity 2006

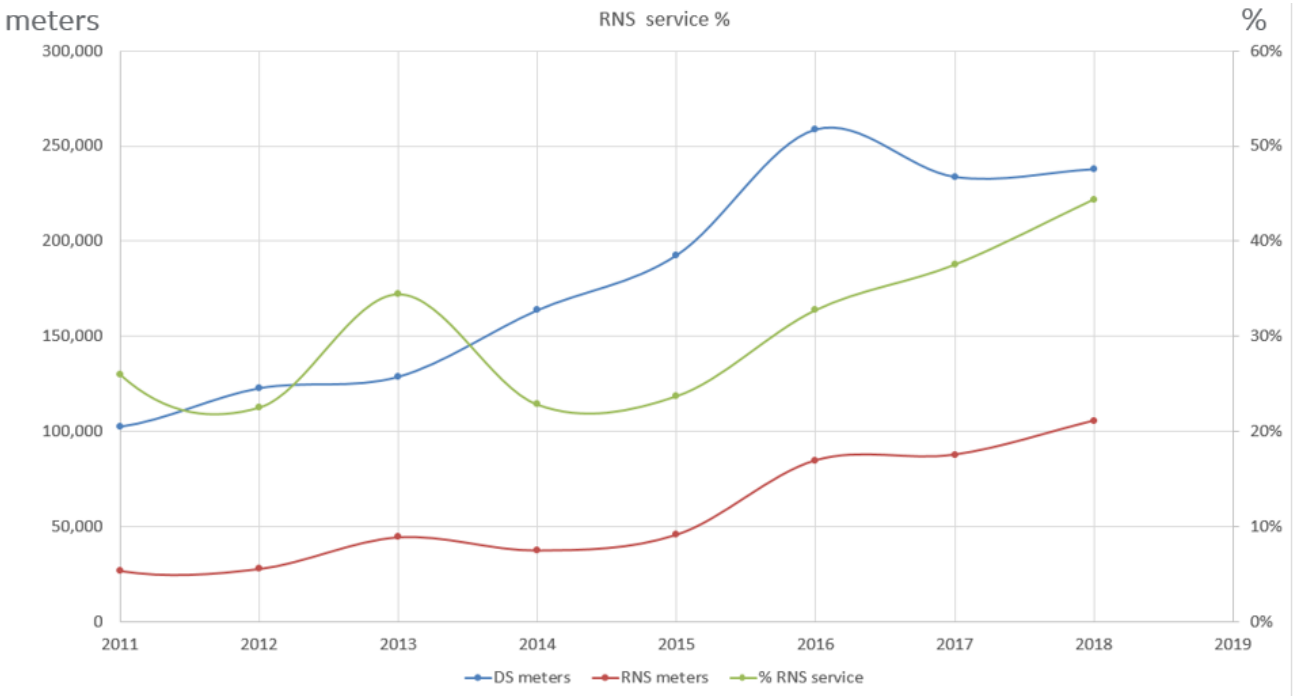
Proper Pro-active geosteering commences



Extra Deep Azimuthal Res. 2012

Opens the doors for proper mapping and analyisis

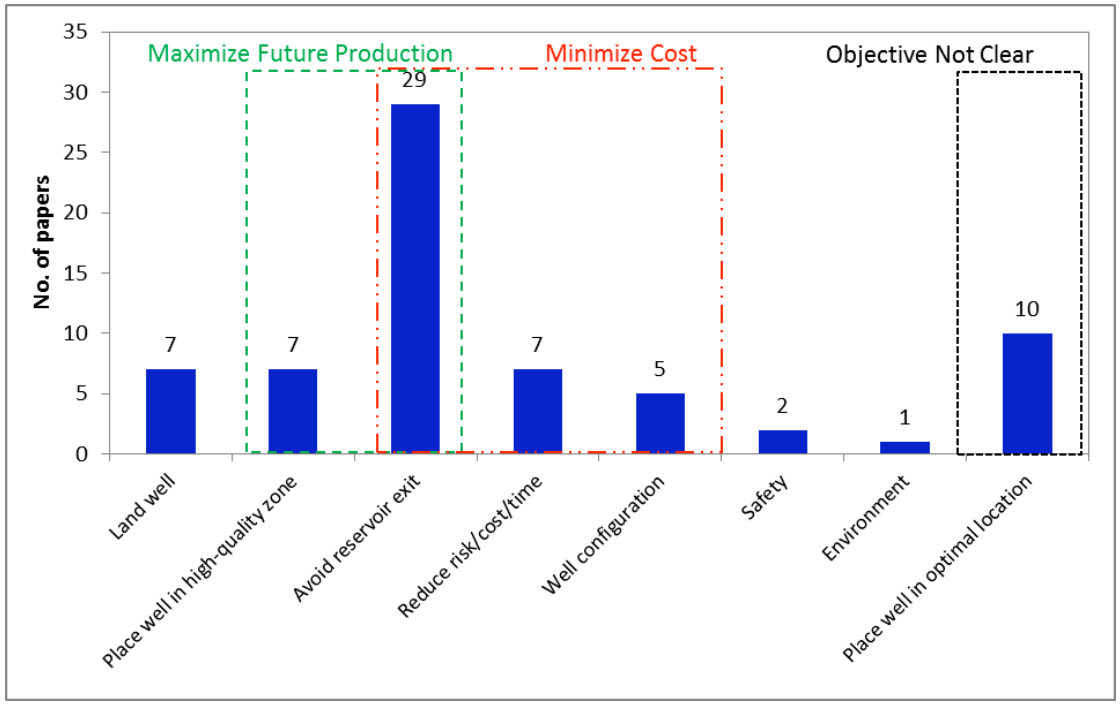




[Data from Baker Hughes, a GE company]

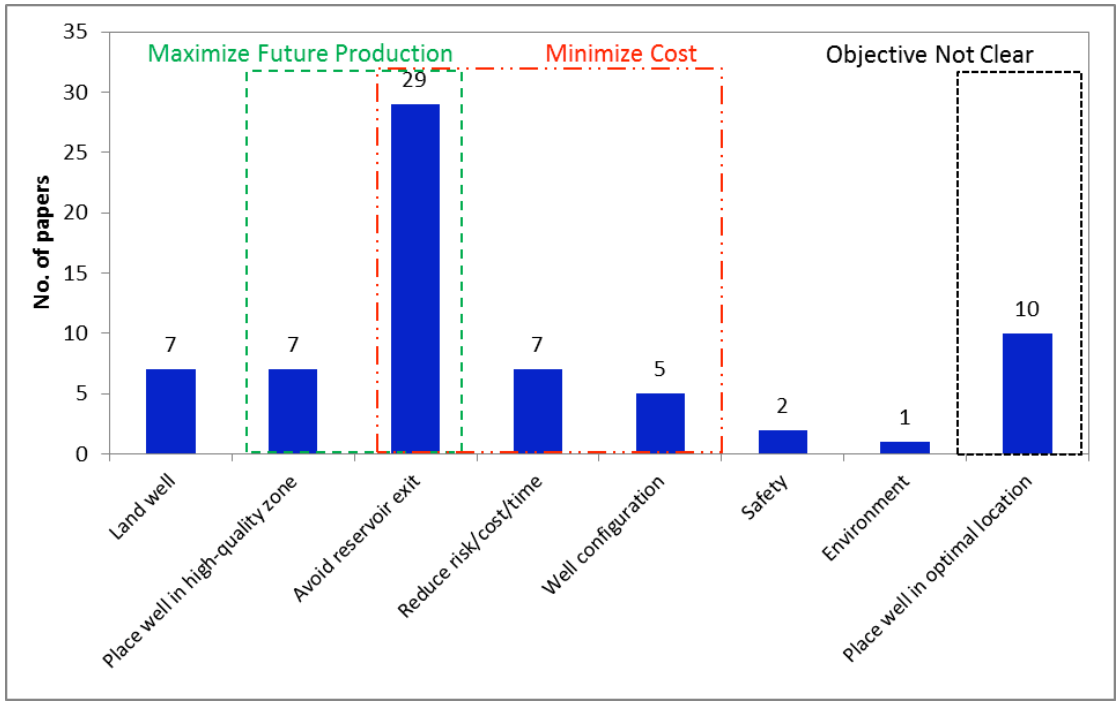
History of geosteering: realtime optimization of well placement (1)

History of geosteering: realtime optimization of well placement (1)



[Kullawan, Bratvold, Bickel (2014) Value...]

History of geosteering: realtime optimization of well placement (1)



[Kullawan, Bratvold, Bickel (2014) Value...]

Realtime optimization workflows

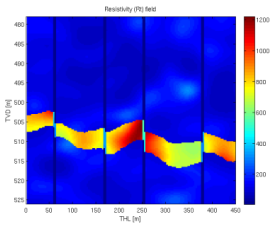
There is a lack of workflows that focus on **systematic optimization of the well placement decisions** while drilling **including uncertainty**.

Representing uncertainty

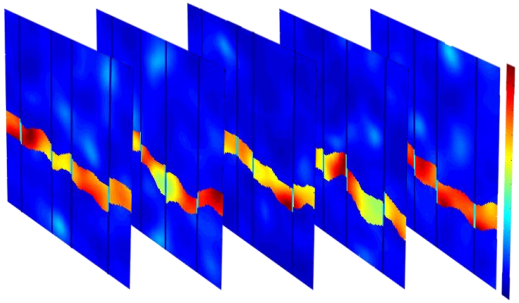
Replace one geomodel (normally used in classical workflows)

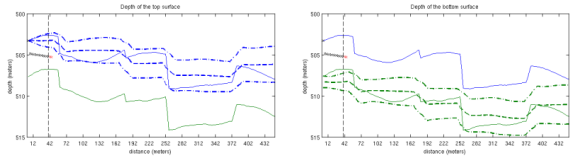
With ensemble of geomodel realizations (representing geological uncertainty)

From which individual uncertainties can be estimated



Resistivity images shown





Uncertain top and bottom boundaries of reservoir layer

Get measurements

Reduce uncertainties

???

Profit

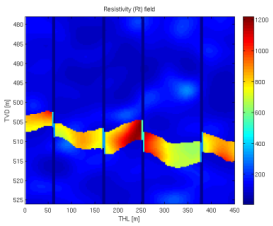
Uncertainties in the problem

Representing uncertainty

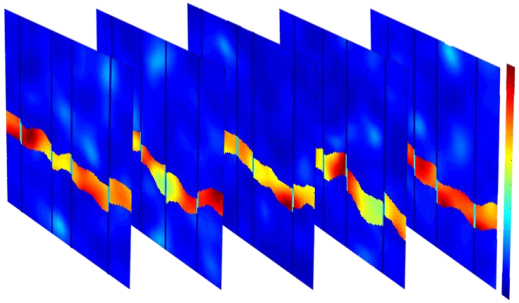
Replace one geomodel (normally used in classical workflows)

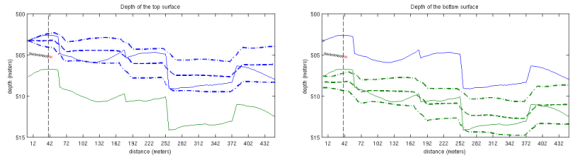
With ensemble of geomodel realizations (representing geological uncertainty)

From which individual uncertainties can be estimated



Resistivity images shown





Uncertain top and bottom boundaries of reservoir layer

Get measurements

Reduce uncertainties

???

Profit

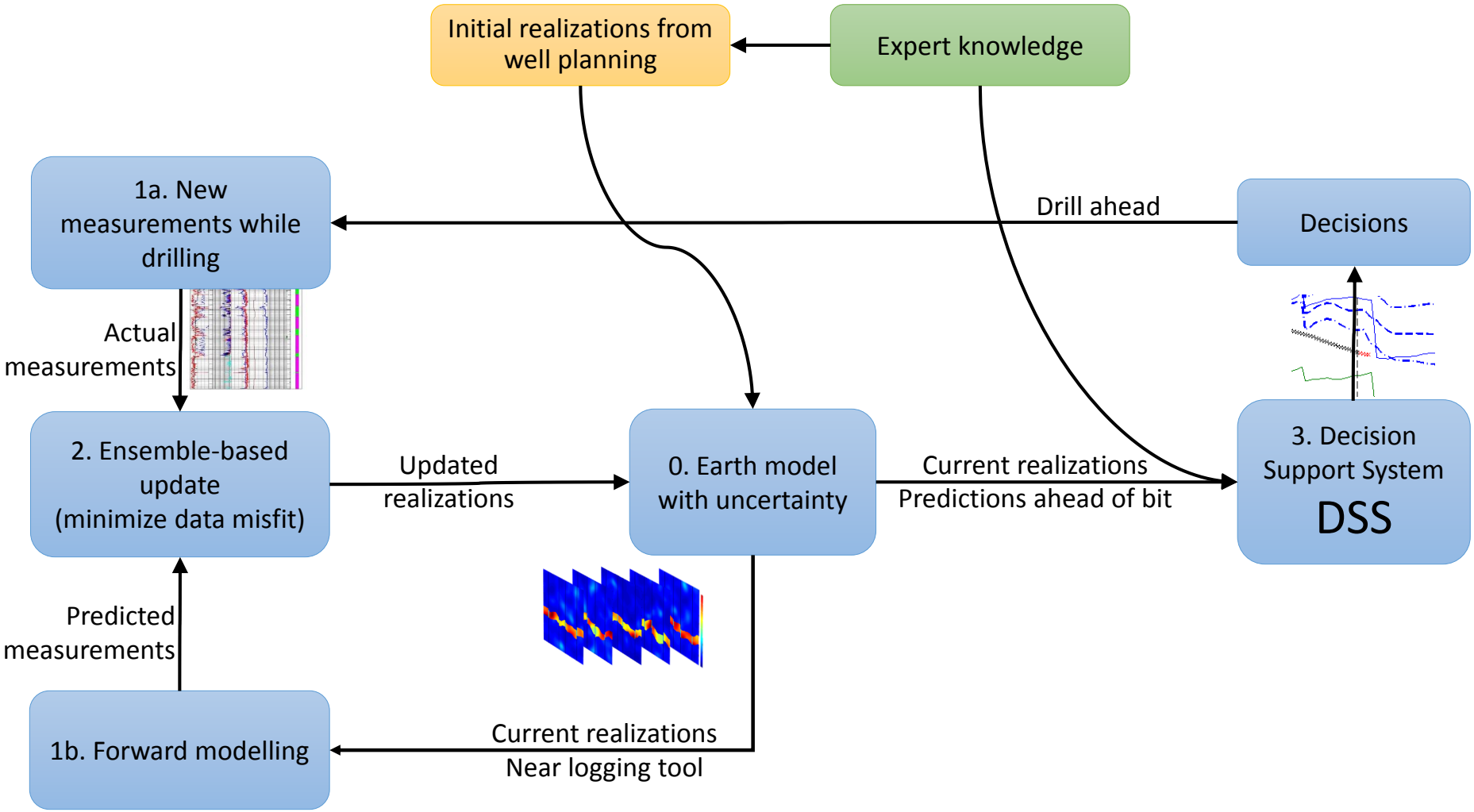
Uncertainties in the problem

Modern reservoir management workflow and expert knowledge*

Initial realizations from well planning

[*Hanea (2015). Reservoir management under geological uncertainty]

Ensemble-based geosteering workflow



[Update workflow: Luo et.al. (2015). An Ensemble-Based Framework...]

Update loop

Ensemble-based update

- Provides incremental update to the uncertain model realizations
- Can work with several types of measurements simultaneously
- Works for any measurement for which we can model

.

Update loop

Ensemble-based update

- Provides incremental update to the uncertain model realizations
- Can work with several types of measurements simultaneously
- Works for any measurement for which we can model

.

The bad news is

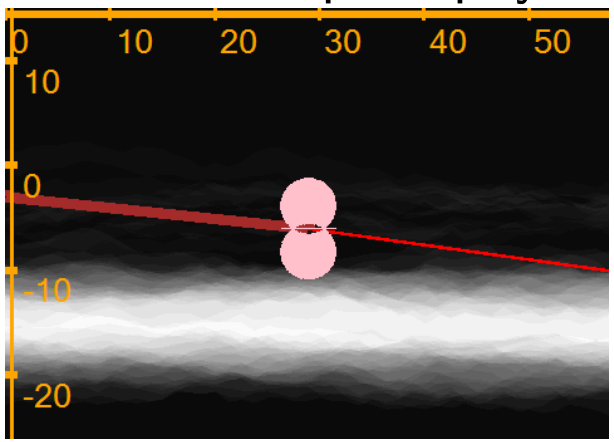
Update loop

Ensemble-based update

- Provides incremental update to the uncertain model realizations
- Can work with several types of measurements simultaneously
- Works for any measurement for which we can model

.

The bad news is
I don't know petrophysics



DOI of our EM tool

The good news is

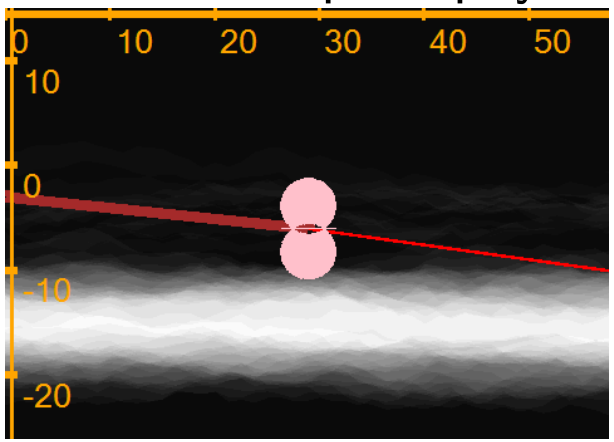
Update loop

Ensemble-based update

- Provides incremental update to the uncertain model realizations
- Can work with several types of measurements simultaneously
- Works for any measurement for which we can model

.

The bad news is
I don't know petrophysics

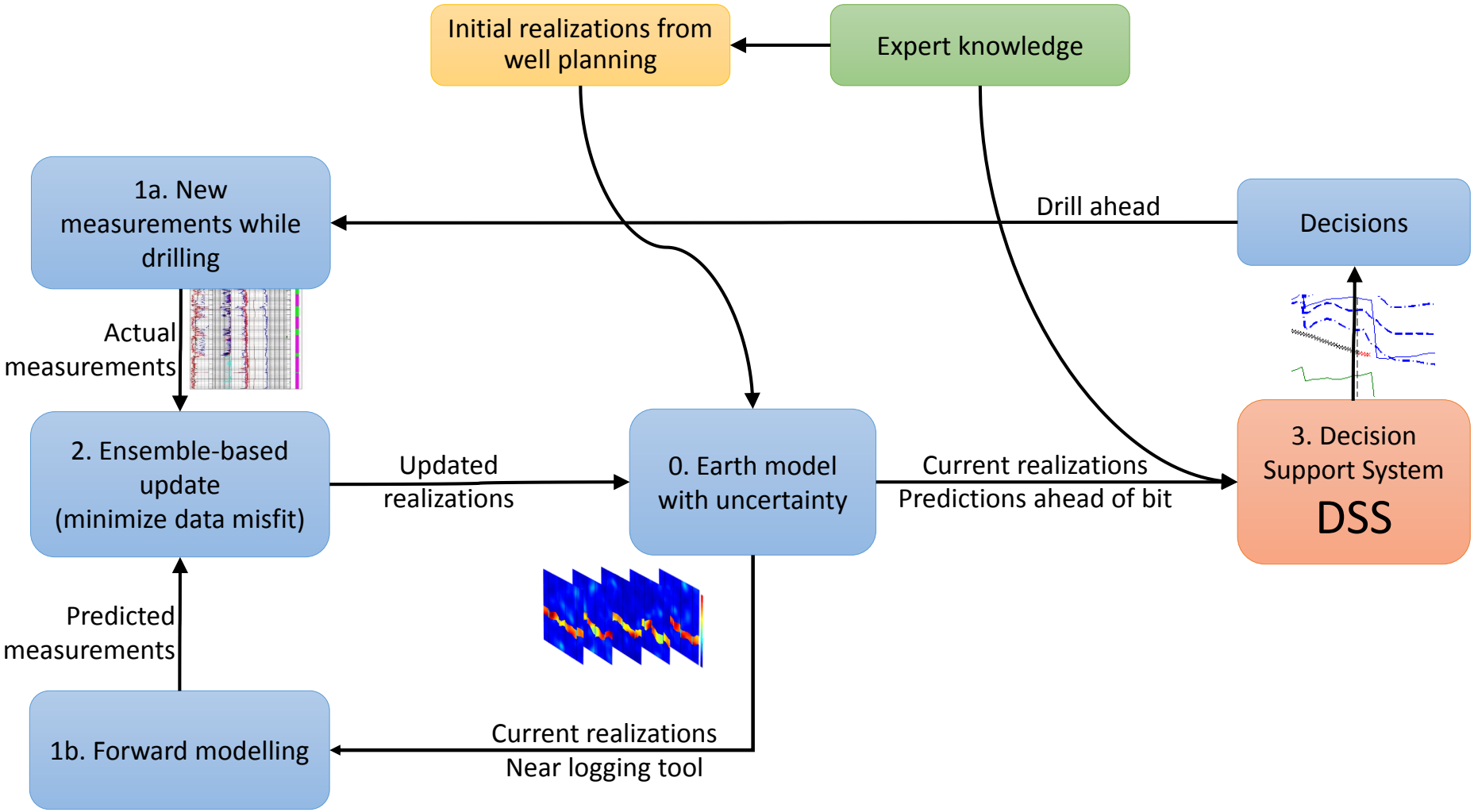


DOI of our EM tool

The good news is
I know a petrophysicst

hope to meet more today

Ensemble-based geosteering workflow



[Update workflow: Luo et.al. (2015). An Ensemble-Based Framework...]

Requirements for a DSS

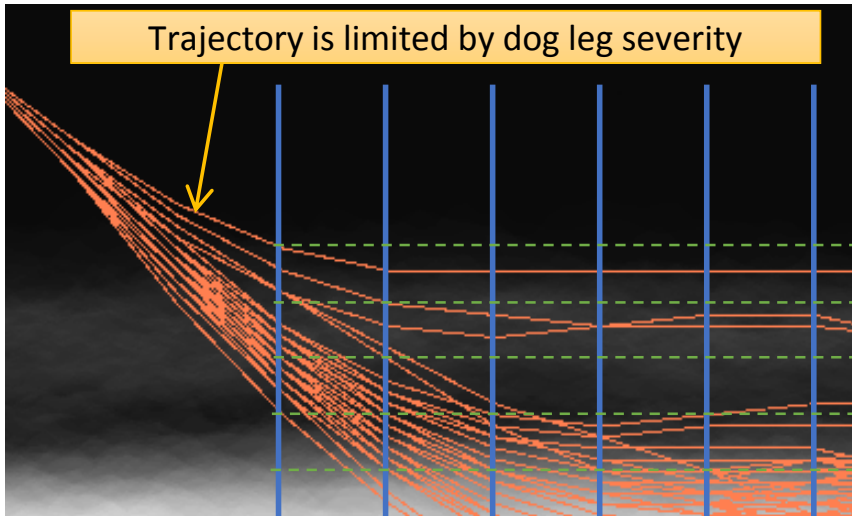
What a DSS can do better than a human?

Requirements for a DSS

What a DSS can do better than a human?

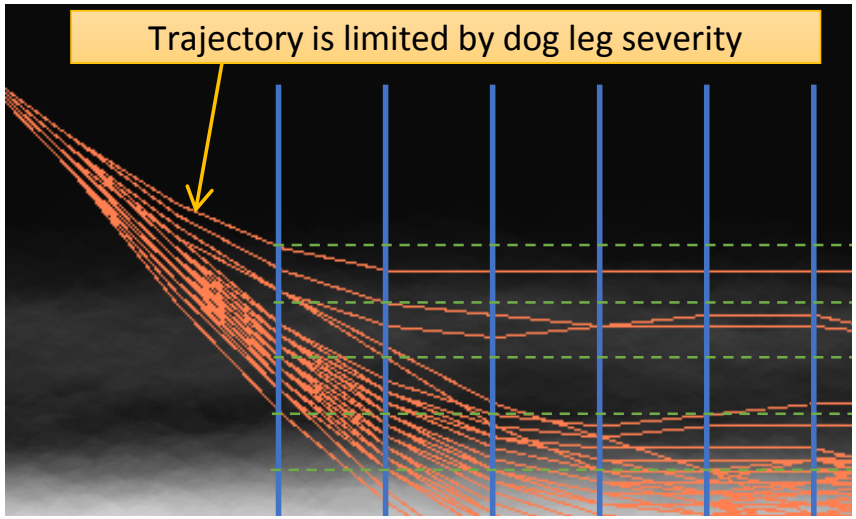
- Realtime performance
- Ability to handle multiple objectives and constraints
- Robust optimization
- Optimality of the decision
  - Optimization of full trajectory ahead of bit
  - Optimality for all objective functions

Algorithm and assumptions



Discretization of trajectories

Algorithm and assumptions



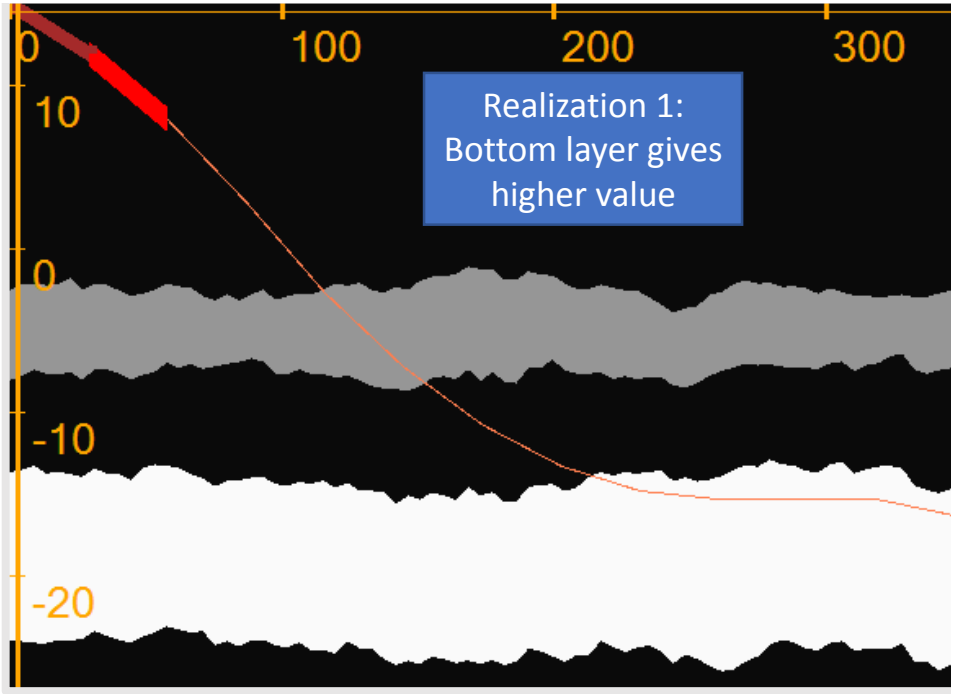
Discretization of trajectories

DSS algorithm: Dynamic Programming

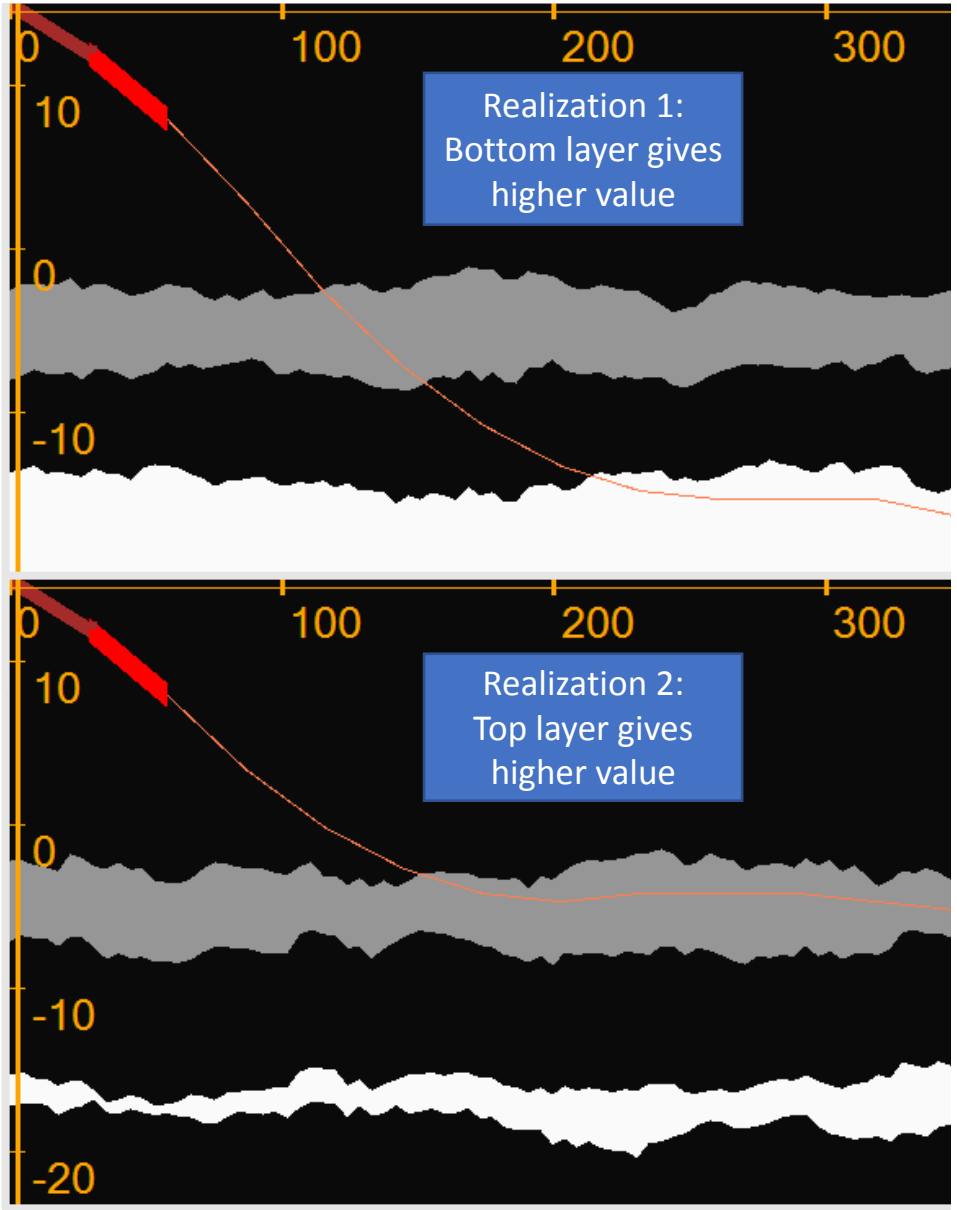
1. Find full best trajectory for every realization and corresponding value
2. Take best decision for the next segment
   - Consider allowed alternatives (continue/steer/stop)
   - Choose best predicted value on average
3. Use new measurements to reduce uncertainty via update loop

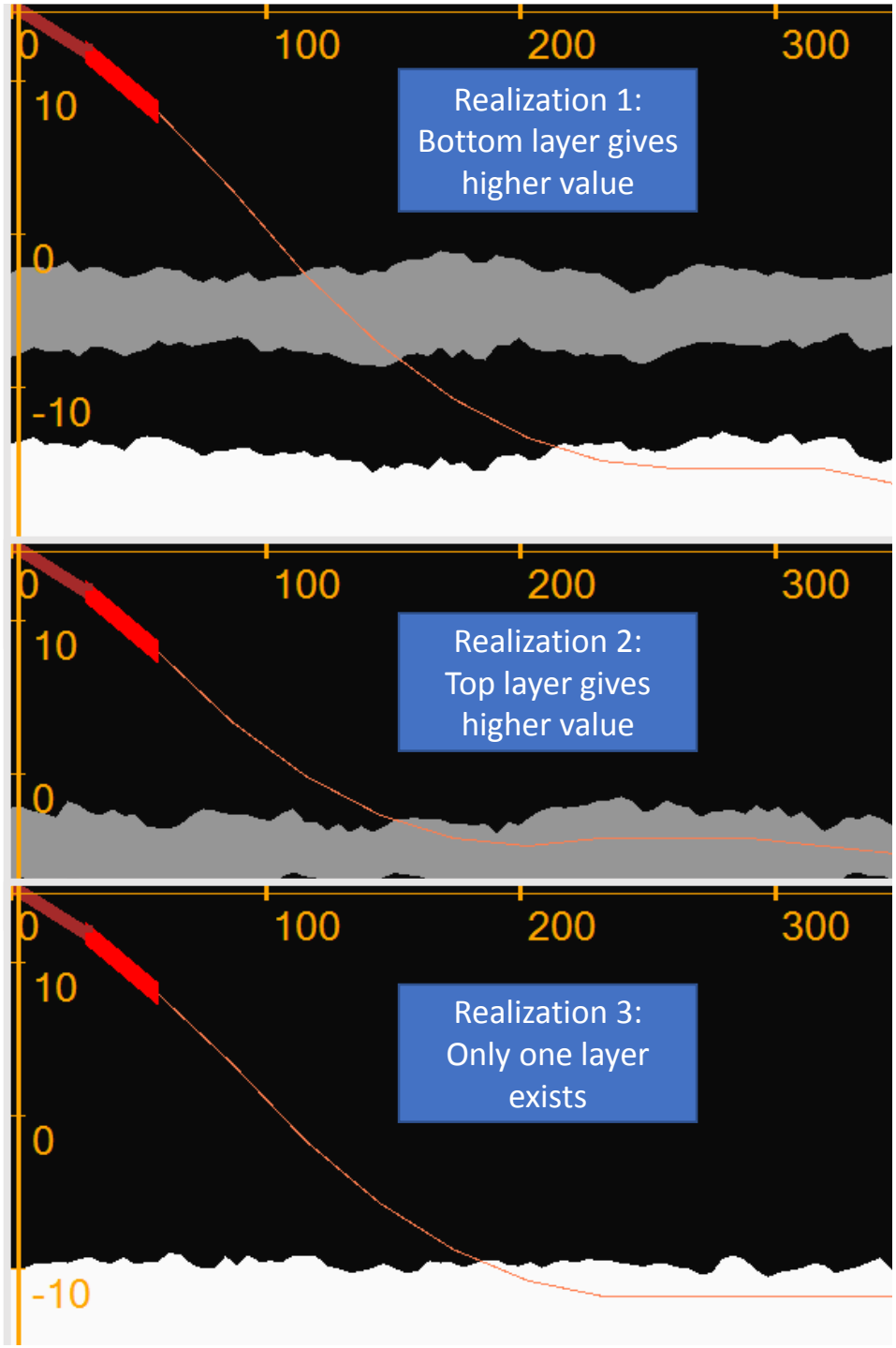
[Inspiration: Kullawan, Bratvold, Bickel (2018). Sequential...]

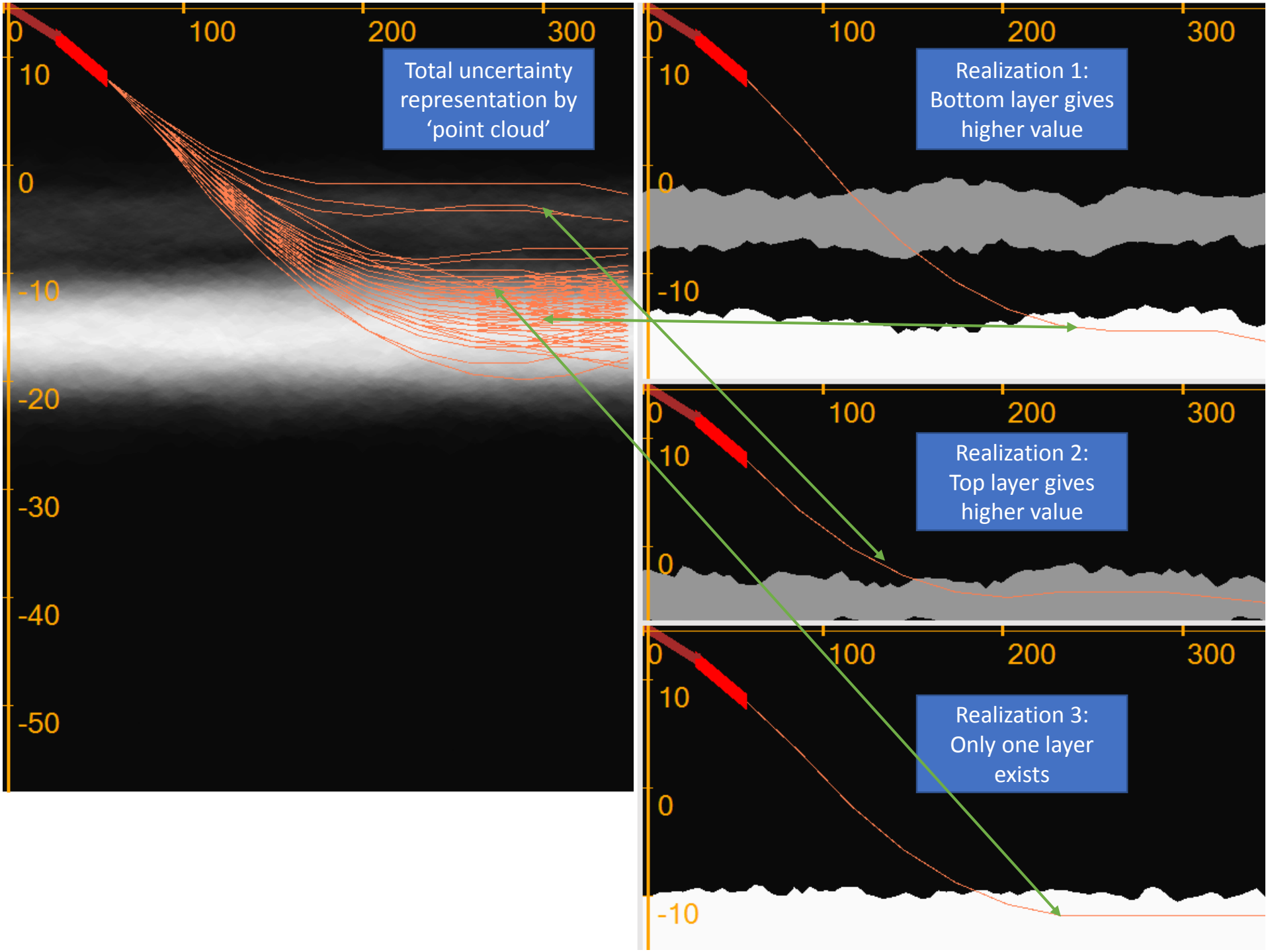
A realizations and its optimal trajectory

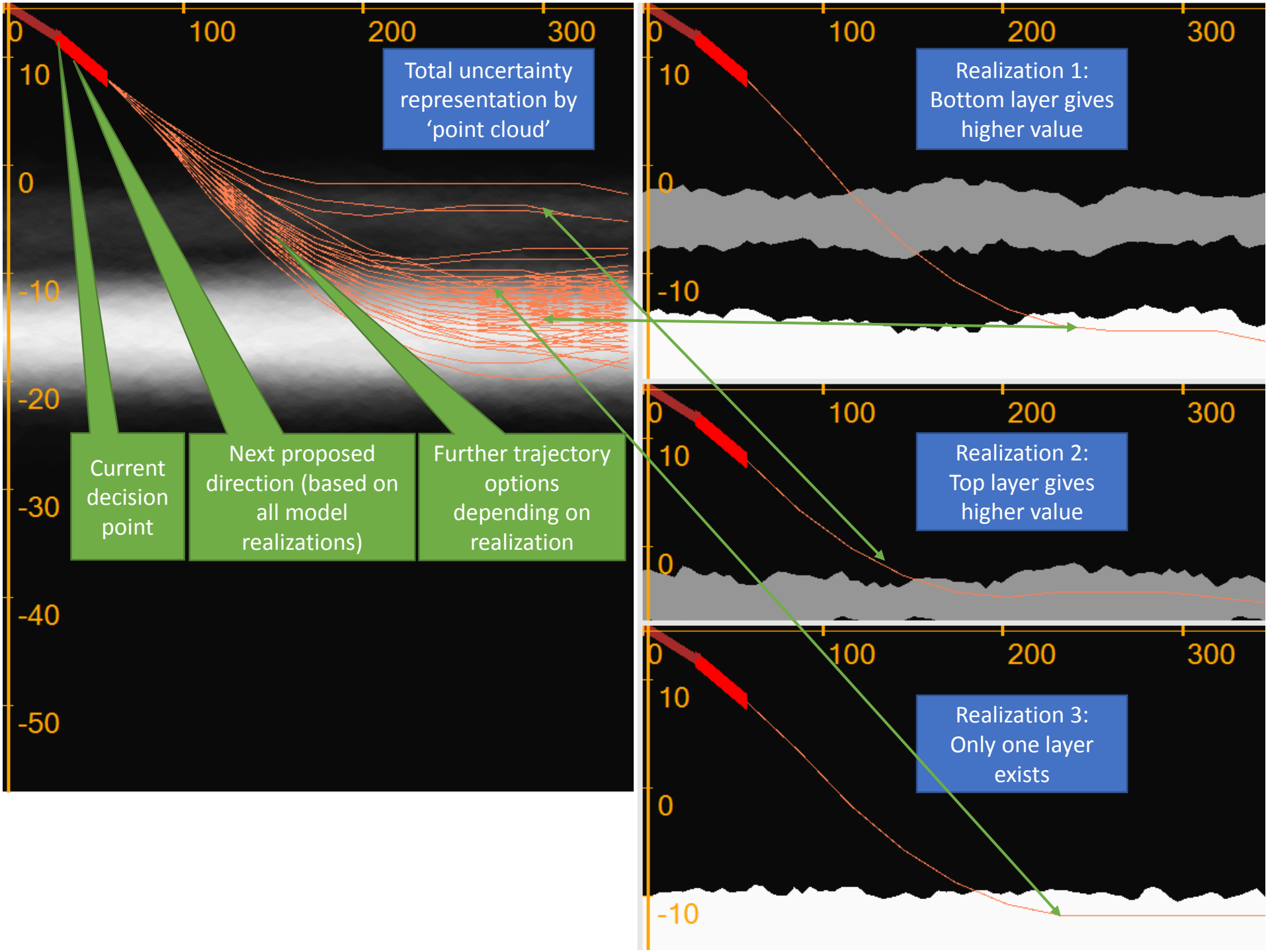


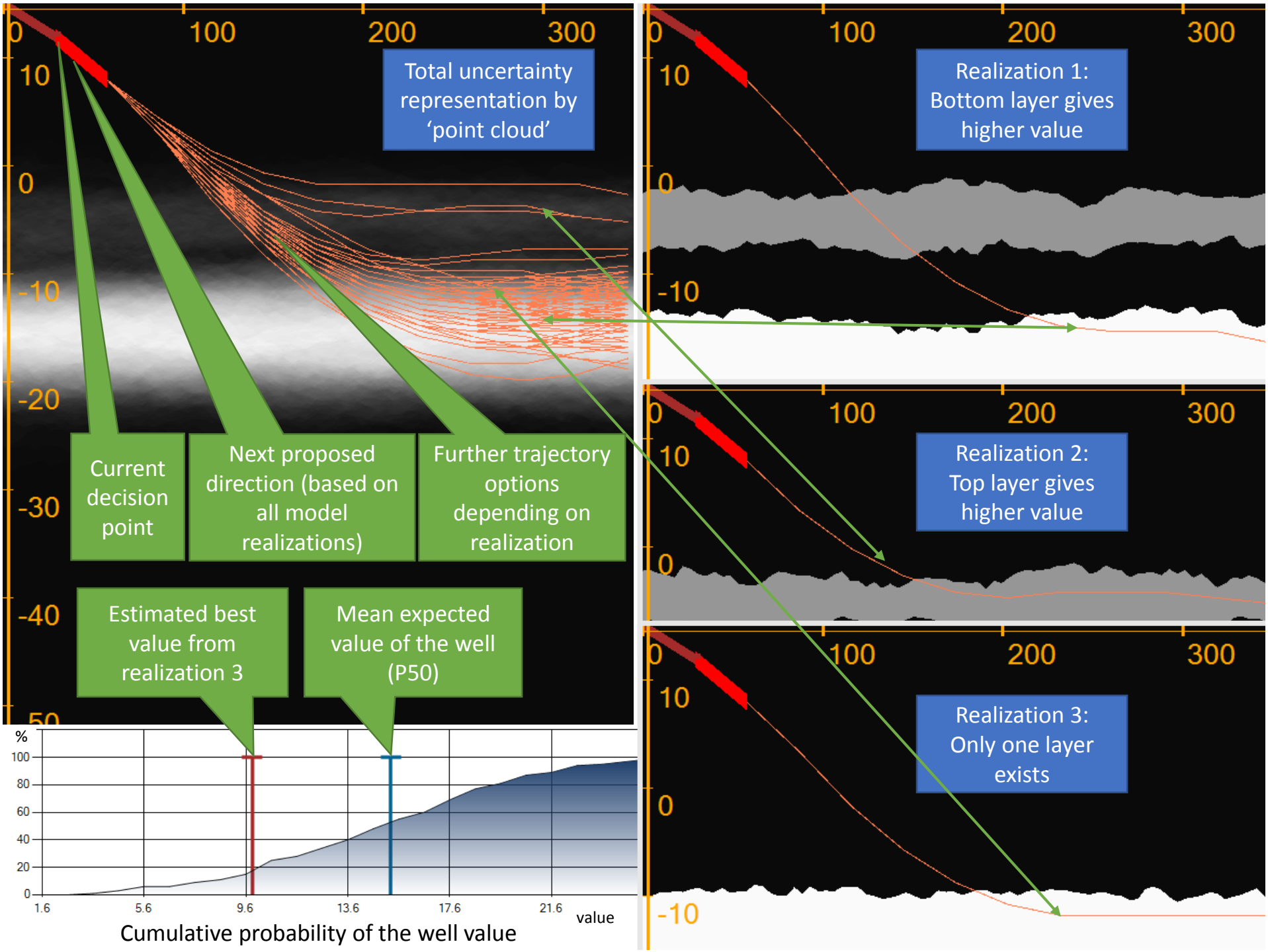
Realizations and optimal trajectories...




0
100
200
300
10
0
-10
Realization 1:
Bottom layer gives
higher value
0
100
200
300
10
0
Realization 2:
Top layer gives
higher value
0
100
200
300
10
0
-10
Realization 3:
Only one layer
exists


Total uncertainty representation by 'point cloud'
Realization 1: Bottom layer gives higher value
Realization 2: Top layer gives higher value
Realization 3: Only one layer exists
0
100
200
300
10
0
-10
-20
-30
-40
-50


0
100
200
300
10
0
-10
-20
-30
-40
-50
Total uncertainty representation by 'point cloud'
Current decision point
Next proposed direction (based on all model realizations)
Further trajectory options depending on realization
Realization 1: Bottom layer gives higher value
Realization 2: Top layer gives higher value
Realization 3: Only one layer exists



Cumulative probability of the well value

Satisfied requirements

DSS algorithm: Dynamic Programming

1. Find full best trajectory for every realization and corresponding value
2. Take best decision for the next segment
   - Consider allowed alternatives (continue/steer/stop)
   - Choose best predicted value on average
3. Use new measurements to reduce uncertainty via update loop

Satisfied requirements

DSS algorithm: Dynamic Programming

1. Find full best trajectory for every realization and corresponding value
2. Take best decision for the next segment
   - Consider allowed alternatives (continue/steer/stop)
   - Choose best predicted value on average
3. Use new measurements to reduce uncertainty via update loop

✓ Robust optimization based on the full ensemble

Satisfied requirements

DSS algorithm: Dynamic Programming

1. Find full best trajectory for every realization and corresponding value
2. Take best decision for the next segment
   - Consider allowed alternatives (continue/steer/stop)
   - Choose best predicted value on average
3. Use new measurements to reduce uncertainty via update loop

✓ Robust optimization based on the full ensemble

✓ Optimality of the decision in the discrete sense

Satisfied requirements

DSS algorithm: Dynamic Programming

1. Find full best trajectory for every realization and corresponding value
2. Take best decision for the next segment
   - Consider allowed alternatives (continue/steer/stop)
   - Choose best predicted value on average
3. Use new measurements to reduce uncertainty via update loop

- ✓ Robust optimization based on the full ensemble
- ✓ Optimality of the decision in the discrete sense
- ✓ Ability to handle multiple objectives and constraints

Satisfied requirements

DSS algorithm: Dynamic Programming

1. Find full best trajectory for every realization and corresponding value
2. Take best decision for the next segment
   - Consider allowed alternatives (continue/steer/stop)
   - Choose best predicted value on average
3. Use new measurements to reduce uncertainty via update loop

- $\surd$ Robust optimization based on the full ensemble
- $\surd$ Optimality of the decision in the discrete sense
- $\surd$ Ability to handle multiple objectives and constraints
- $*$ Realtime performance

$\surd$ - by construction.

Example 1: optimal landing — setup

Value function measured in "equivalent meters of reservoir":

- reservoir thickness when drilling in the reservoir
- the value is doubled in the 'sweet spot' between 0.75 and 2.25 meters from the reservoir top
- a pre-set cost per meter of well.

Constraints:

- Max dogleg severity 2 deg.
- Max inclination 90 deg from vertical.

Example 1: optimal landing — setup

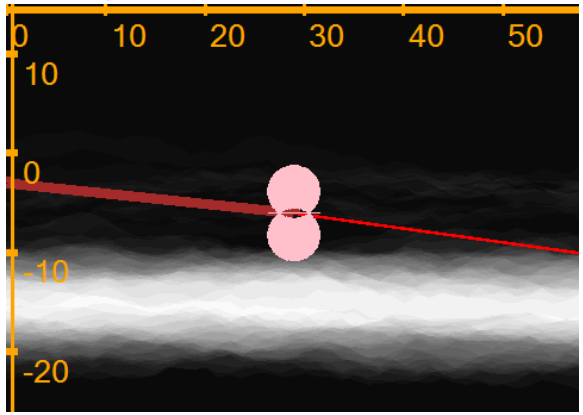
Value function measured in "equivalent meters of reservoir":

- reservoir thickness when drilling in the reservoir
- the value is doubled in the 'sweet spot' between 0.75 and 2.25 meters from the reservoir top
- a pre-set cost per meter of well.

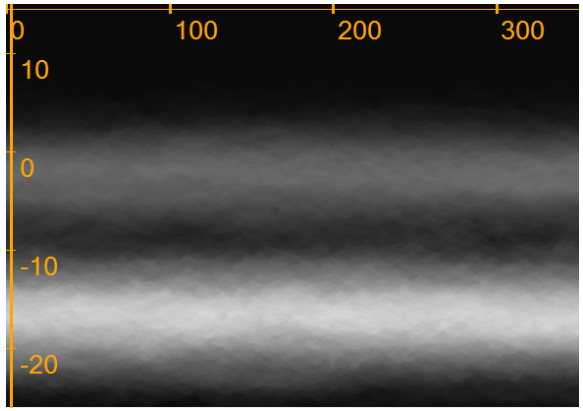
Constraints:

- Max dogleg severity 2 deg.
- Max inclination 90 deg from vertical.

DOI of our EM tool



Initial Ensemble:

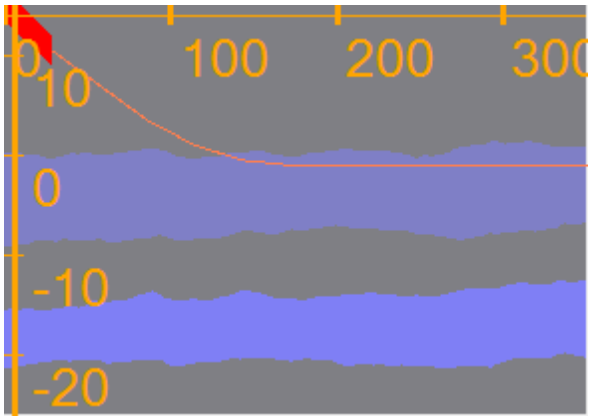


Expected: 2 reservoir layers and background shales uncertain boundaries

Example 1: optimal landing

Synthetic truth

Case A:

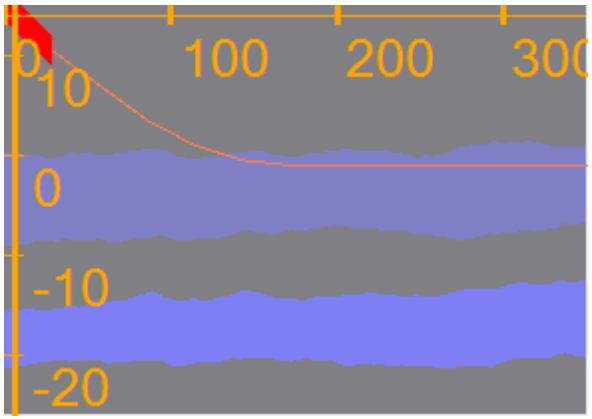


The truth that is statistically expected

Example 1: optimal landing

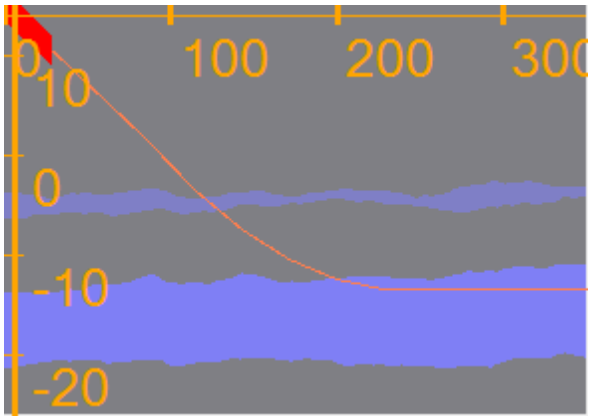
Synthetic truth

Case A:



The truth that is statistically expected

Case B:



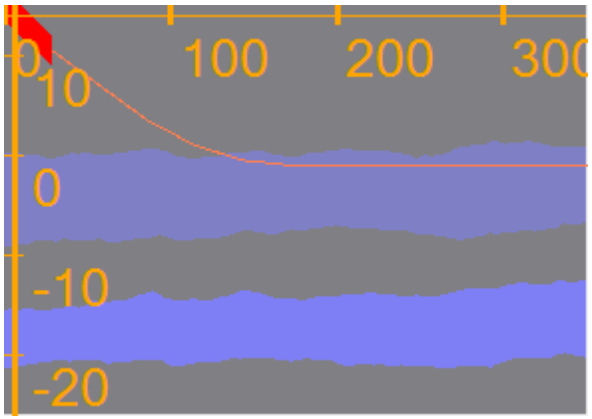
Truth with a degenerate top layer that differs from expectation

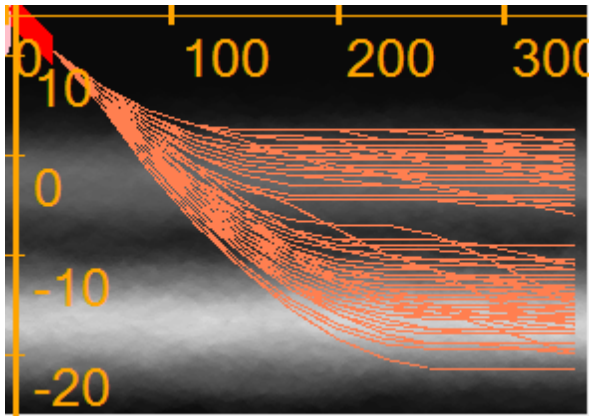
Example 1: optimal landing — two scenarios

Synthetic truth

Step 0

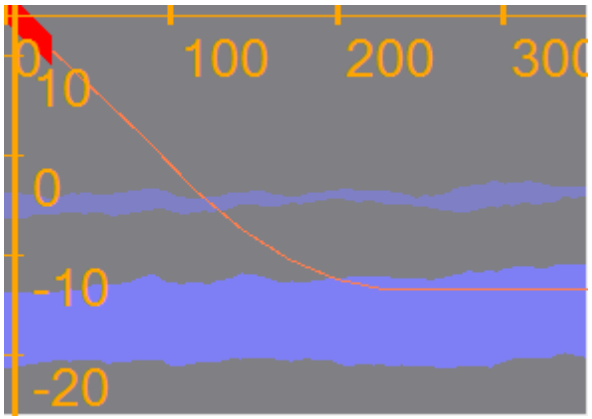
A

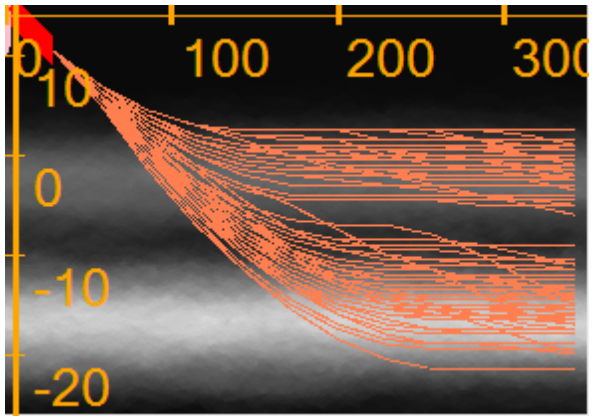




Expect two layers of good sands

B





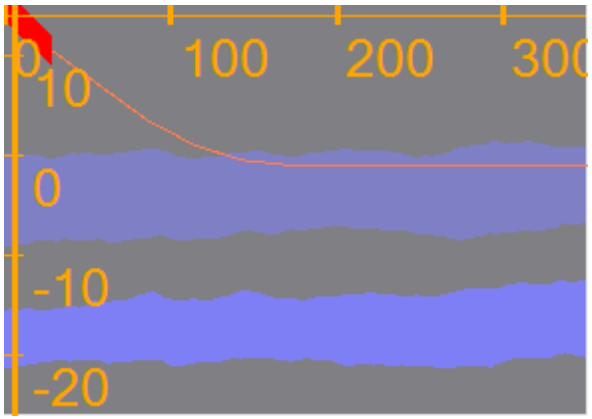
Expect two layers of good sands

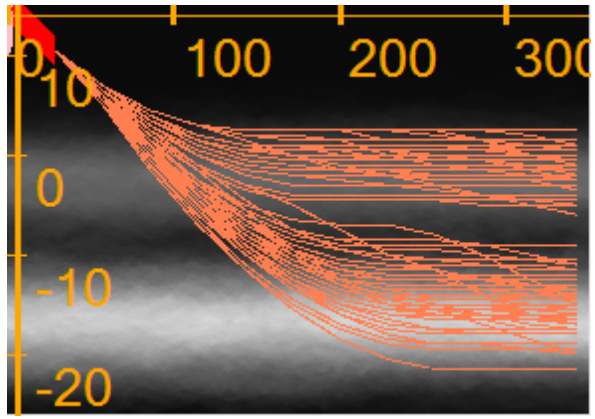
Identical setup

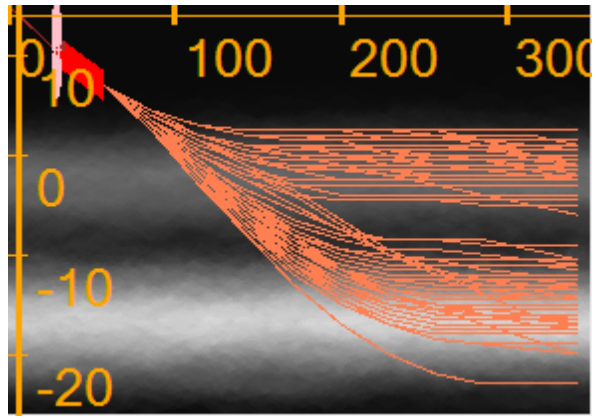
Example 1: optimal landing — two scenarios

Synthetic truth | Step 0 | Step 1

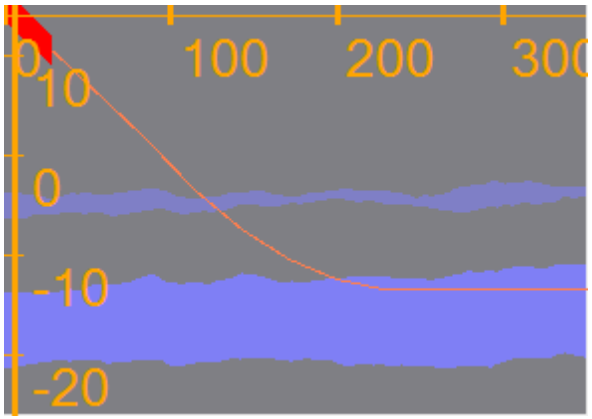
A

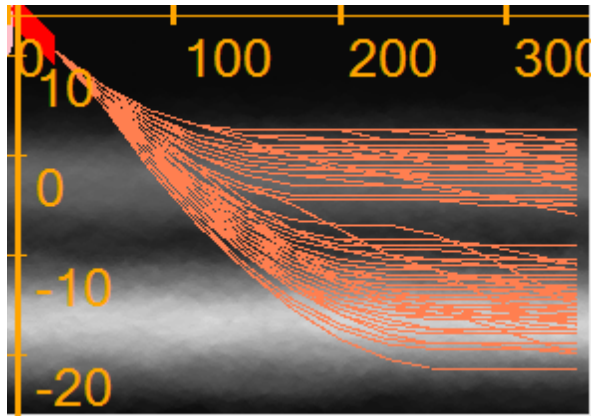


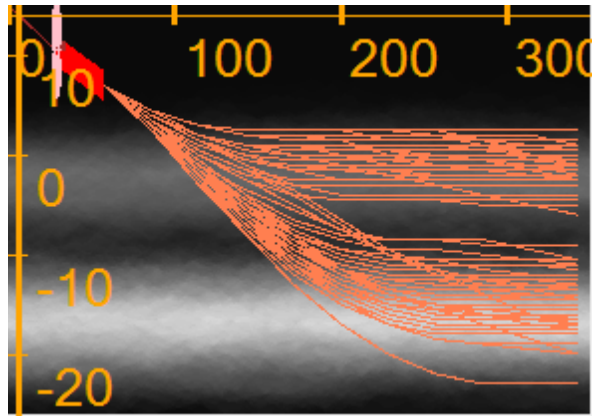




B







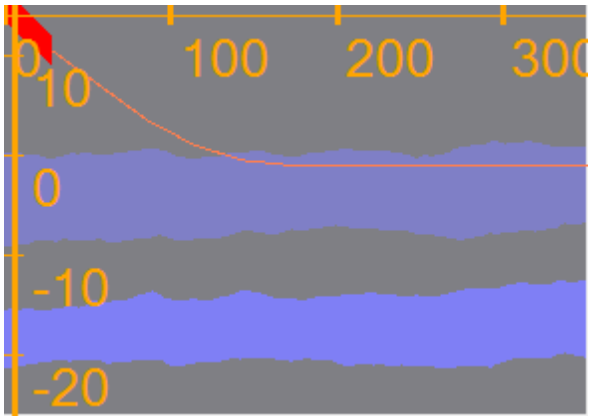
Identical setup

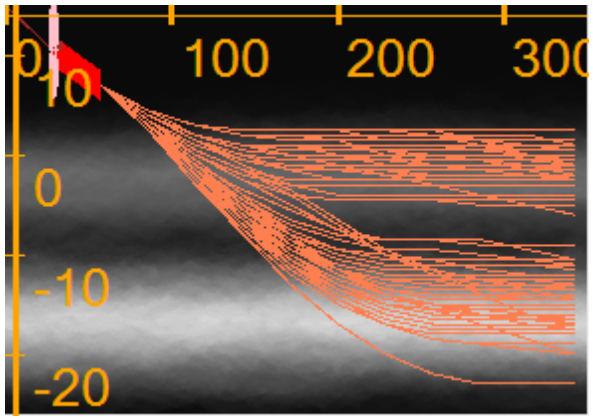
No update

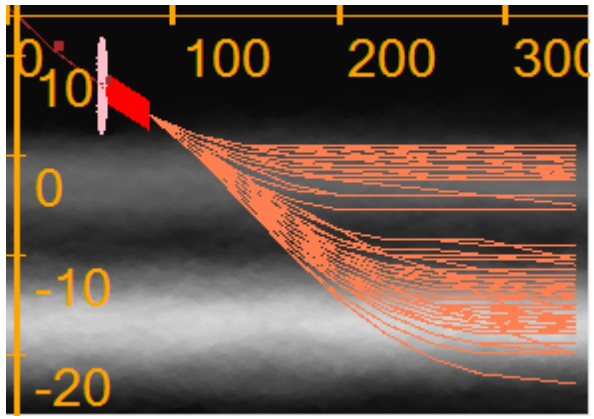
Example 1: optimal landing — two scenarios

Synthetic truth | Step 1 | Step 2

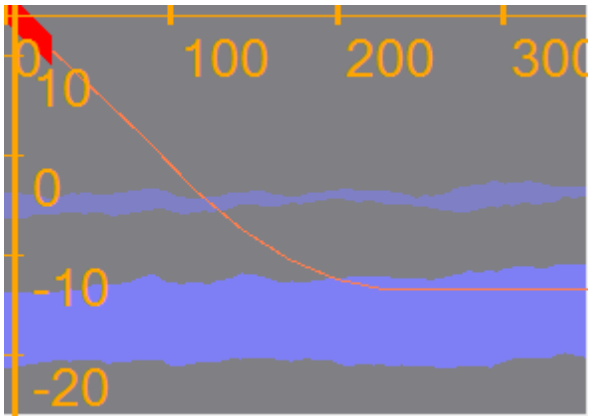
A

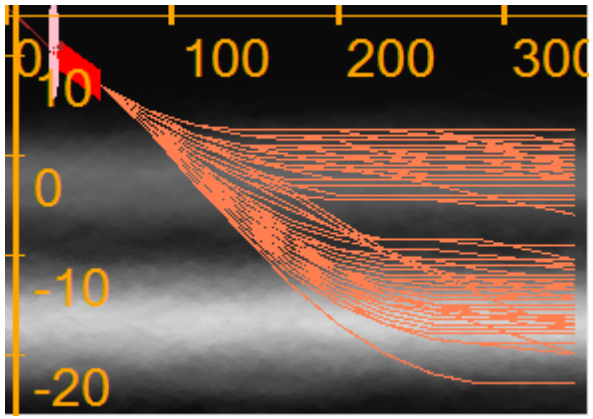


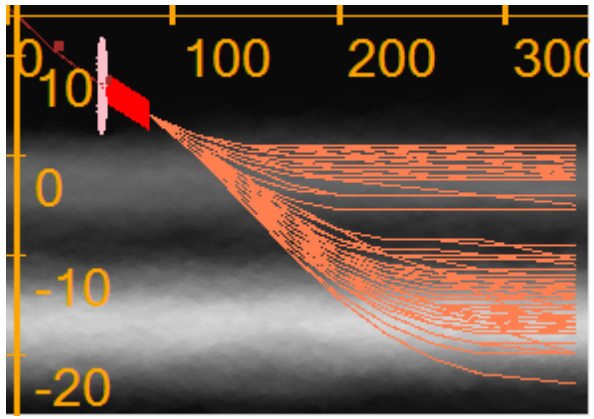




B

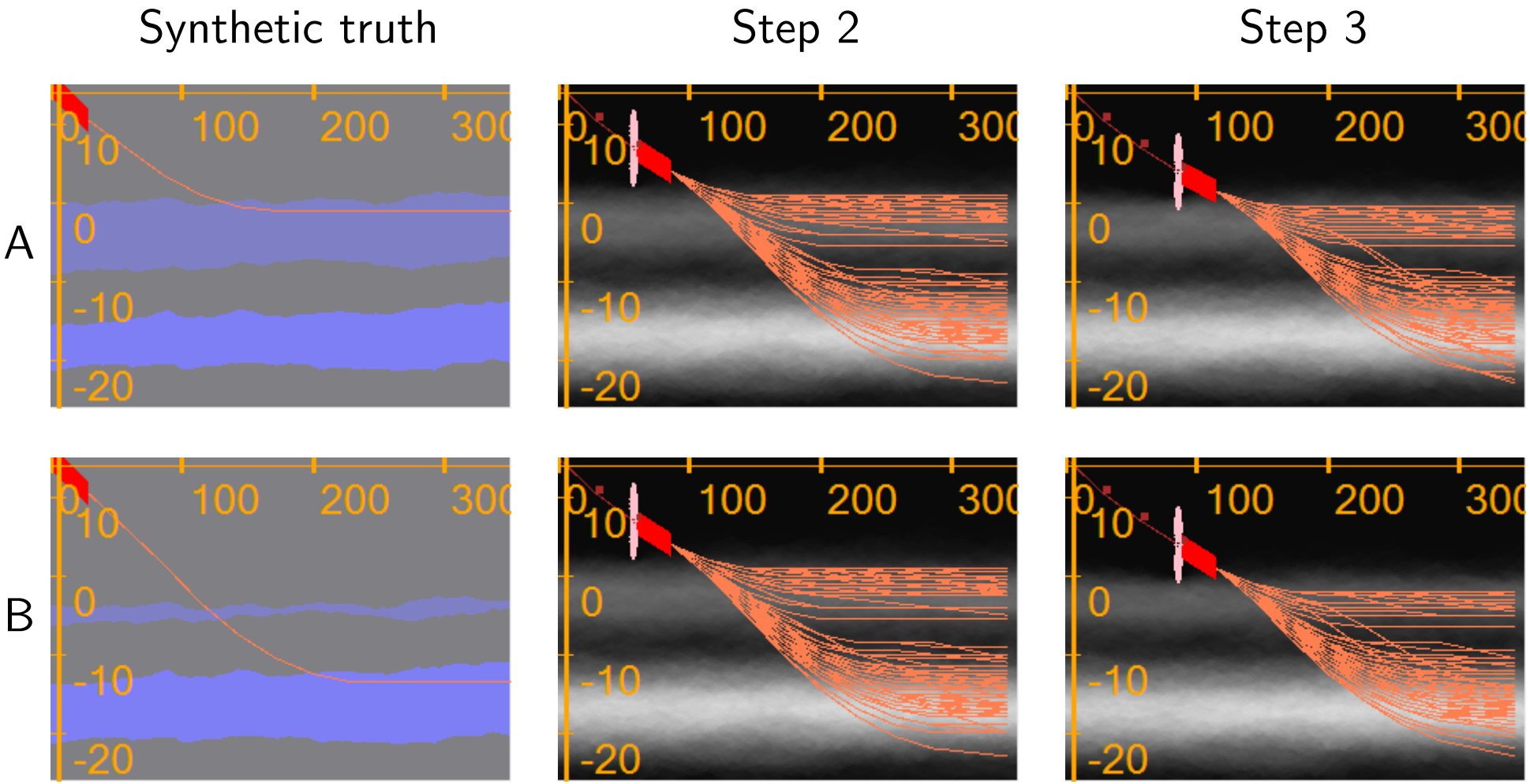






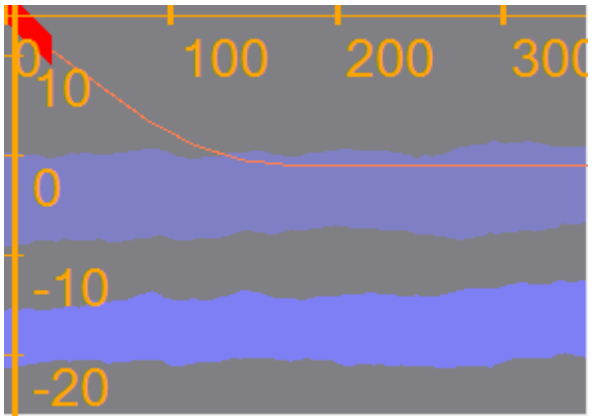
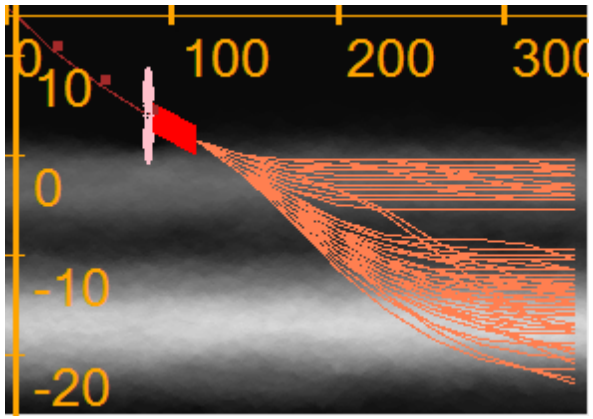
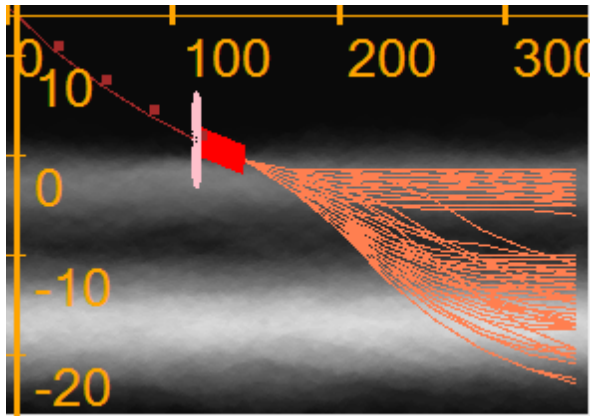
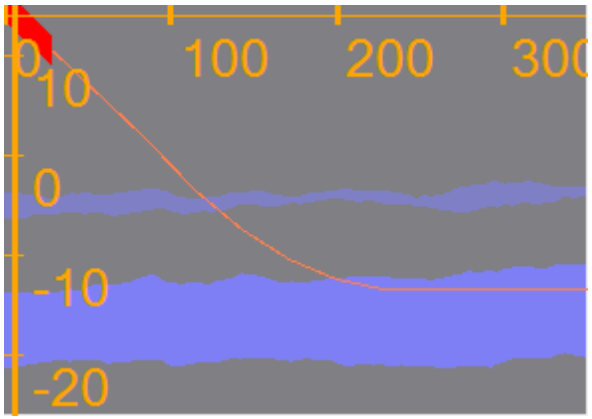
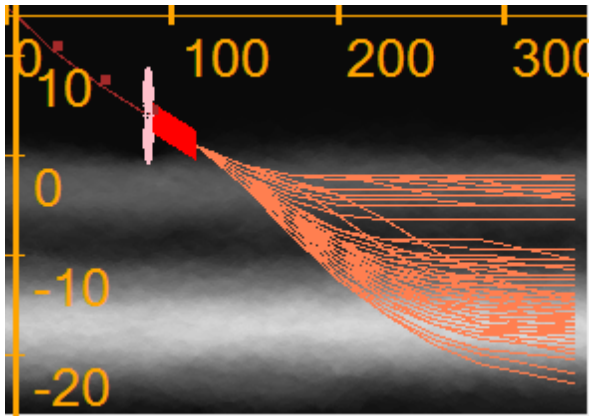
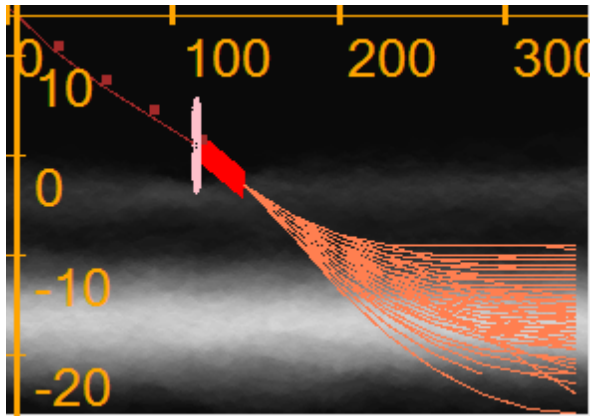
No update

Example 1: optimal landing — two scenarios



Look-around touches expected boundary

Example 1: optimal landing — two scenarios

	Synthetic truth	Step 3	Step 4
A			
B			

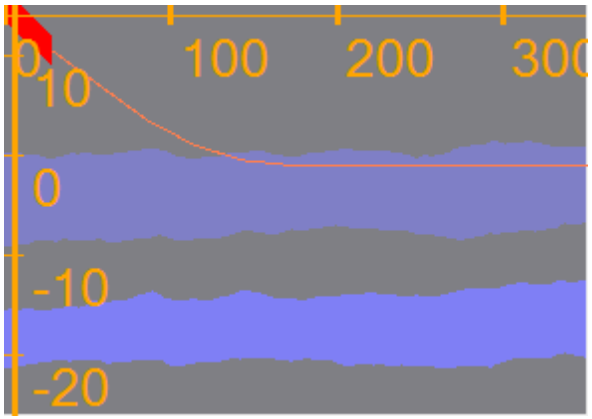
Look-around touches expected boundary

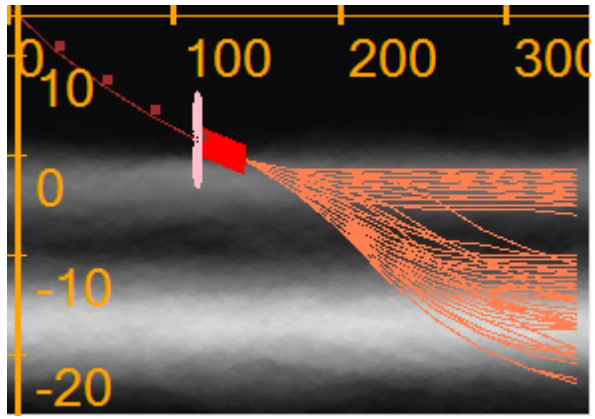
No expected top in B

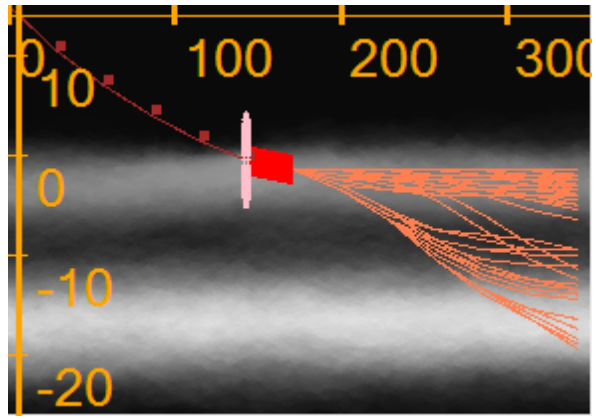
Example 1: optimal landing — two scenarios

Synthetic truth | Step 4 | Step 5

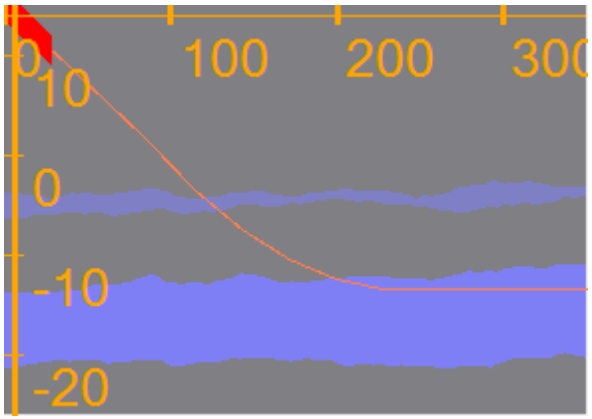
A

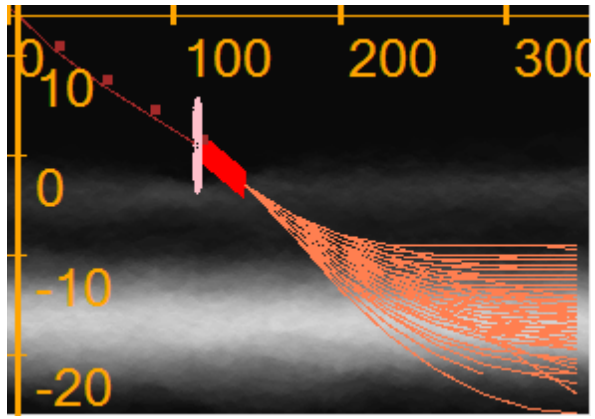


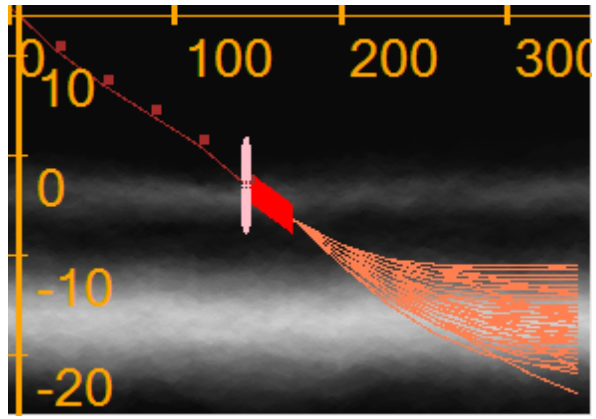




B







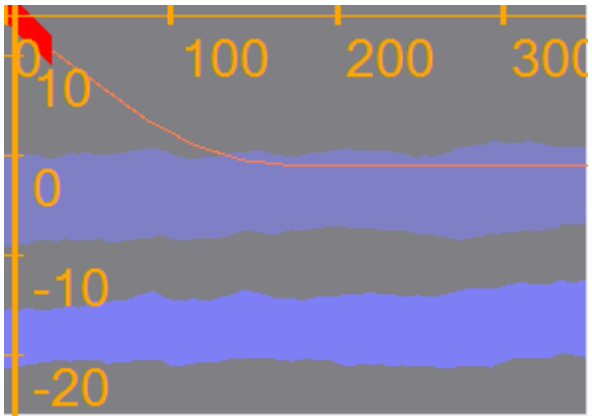
No expected top in B

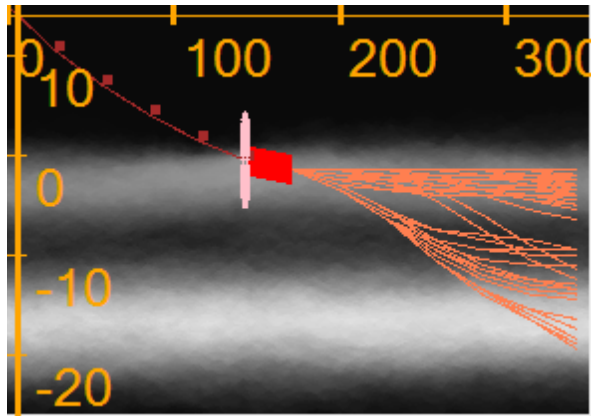
In A bottom layer seems better for some realizations

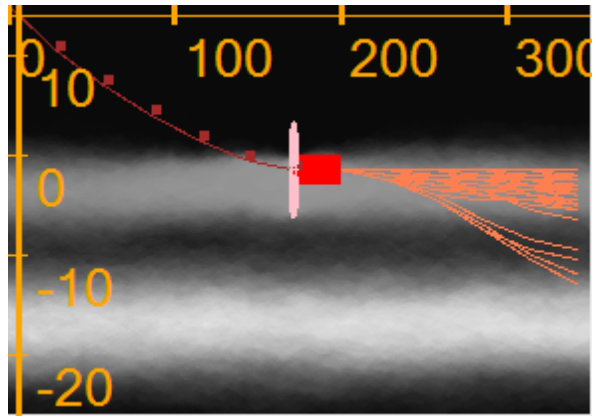
Example 1: optimal landing — two scenarios

Synthetic truth | Step 5 | Step 6

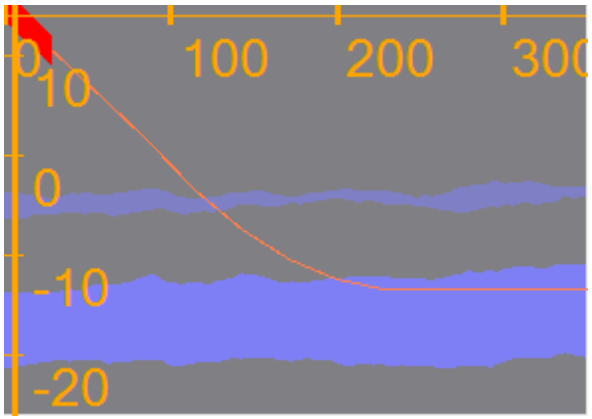
A

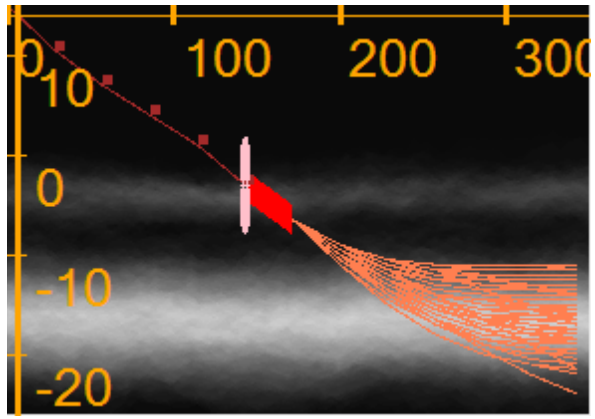


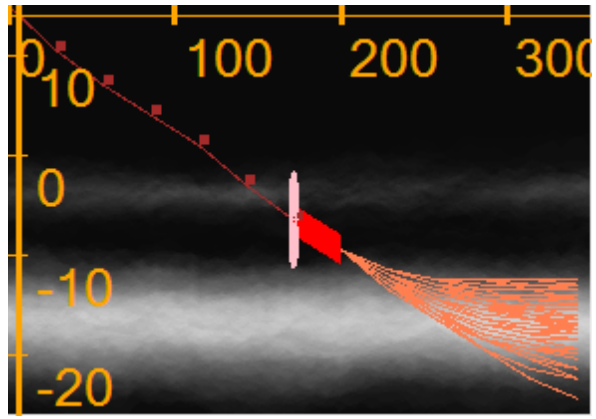




B

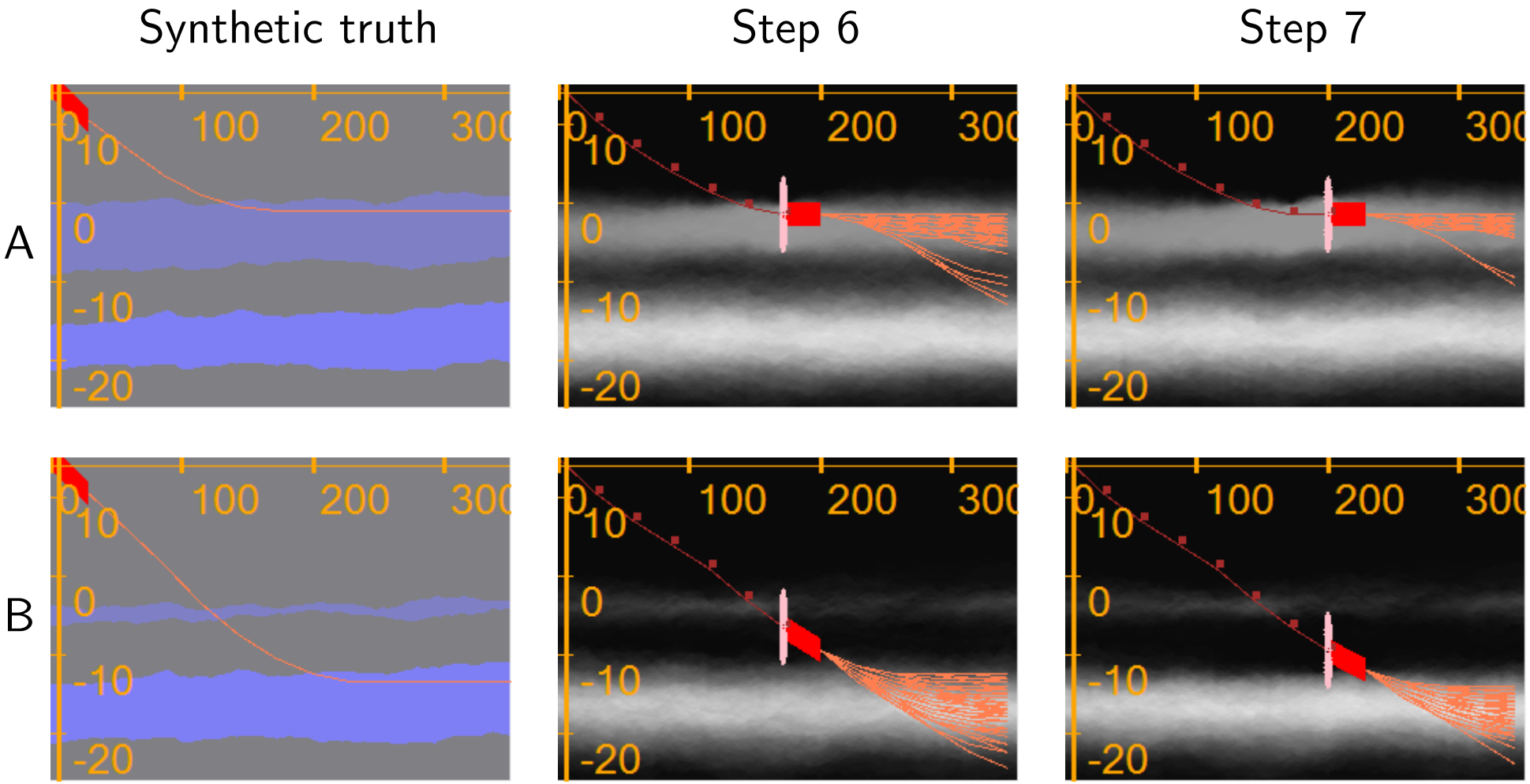






In A bottom layer seems better for some realizations

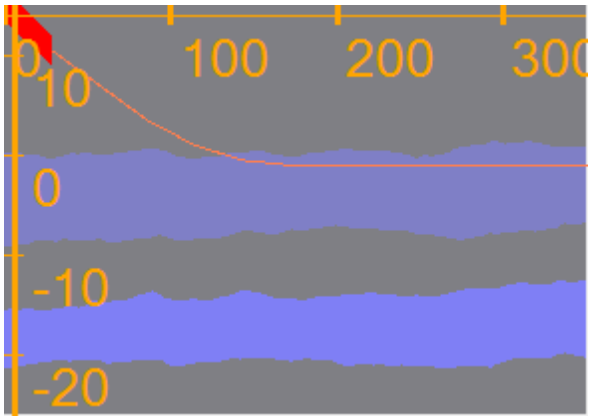
Example 1: optimal landing — two scenarios

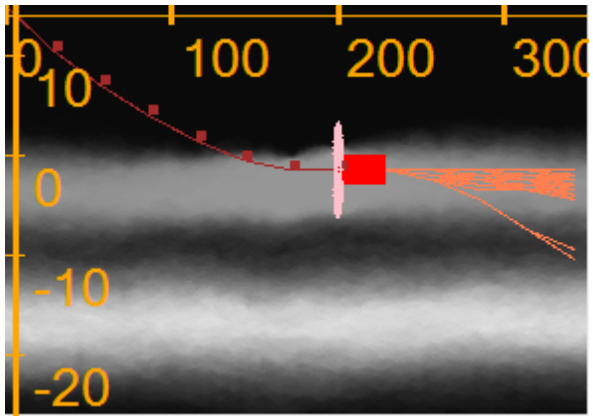


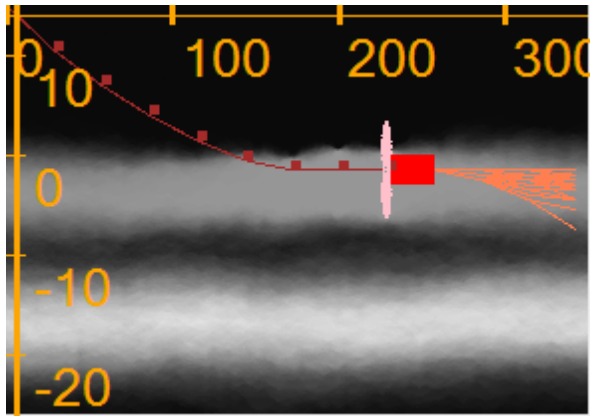
Example 1: optimal landing — two scenarios

Synthetic truth | Step 7 | Step 8

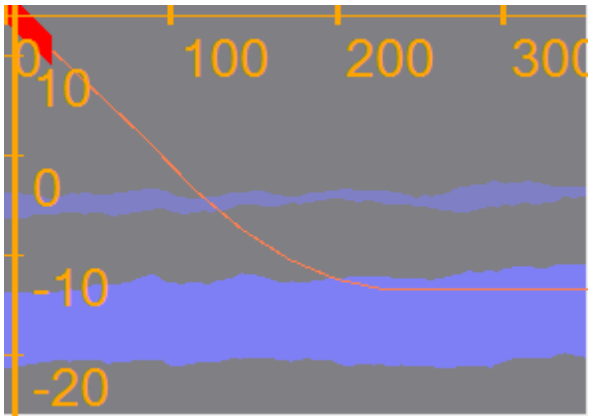
A

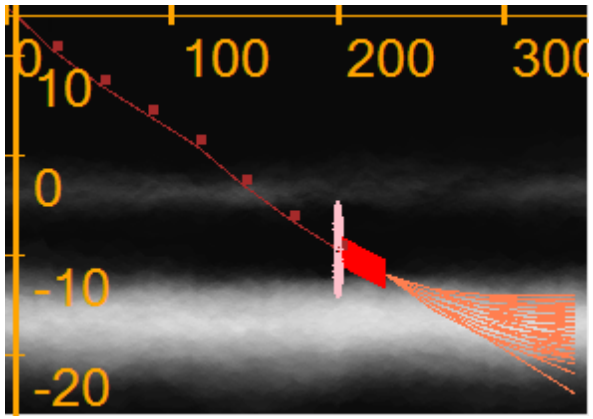


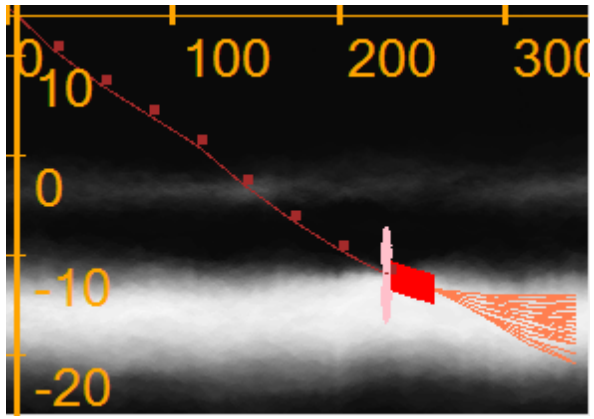




B







All realizations follow 'correct' layer

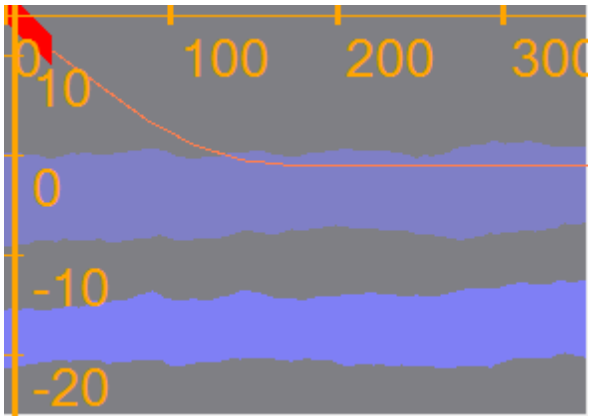
Example 1: optimal landing — two scenarios

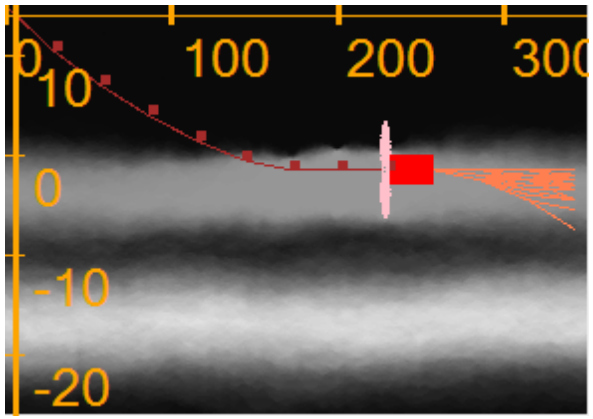
Synthetic truth

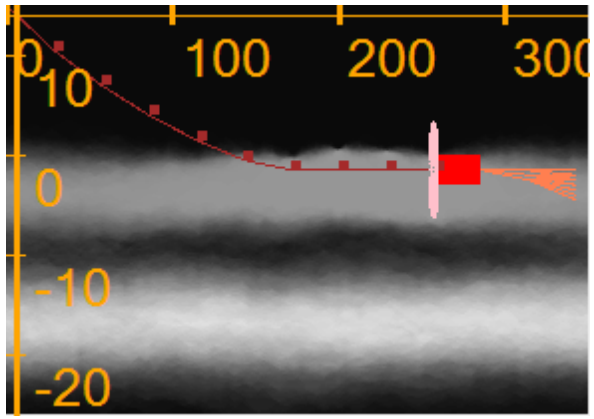
Step 8

Step 9

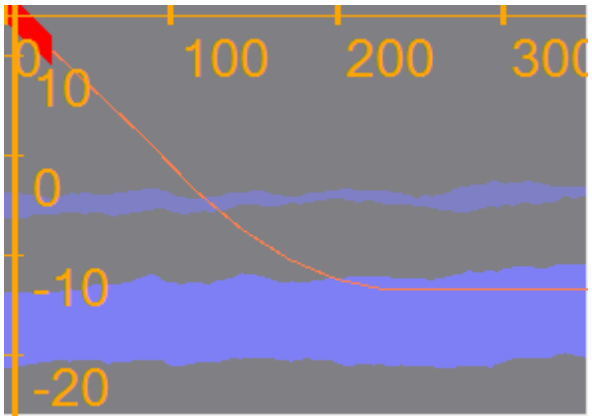
A

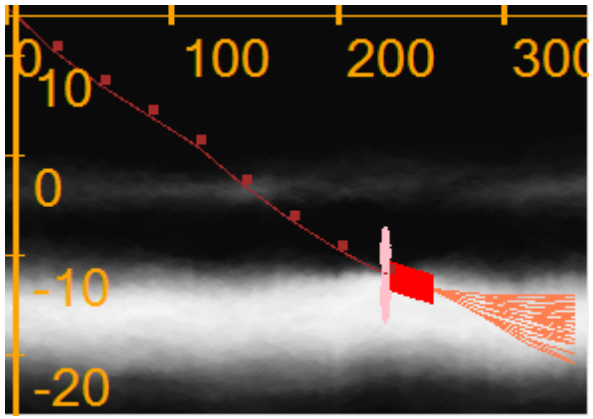


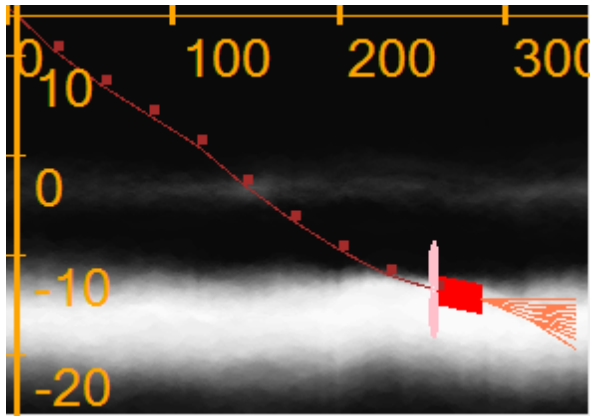




B





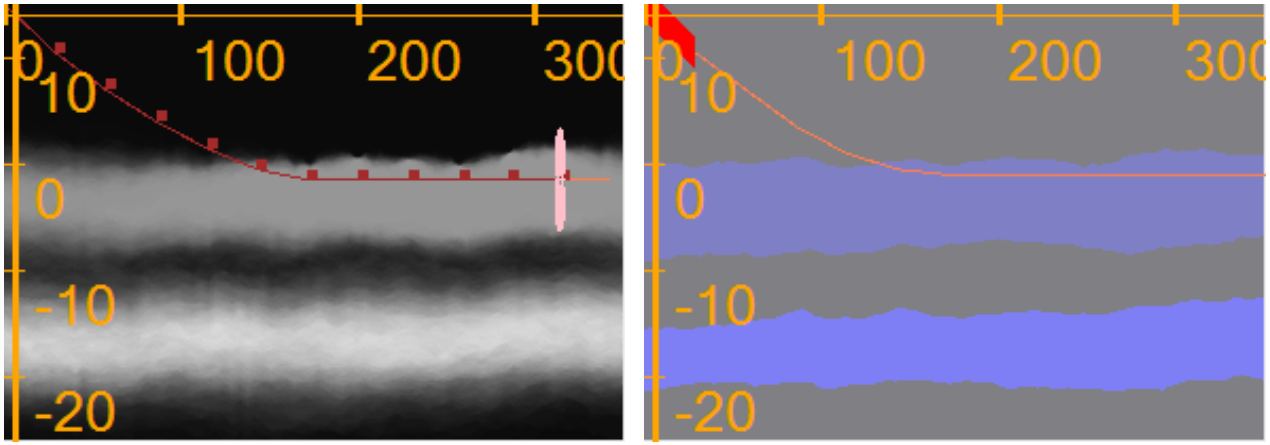


All realizations follow 'correct' layer

Final stage

Example 1: Final state

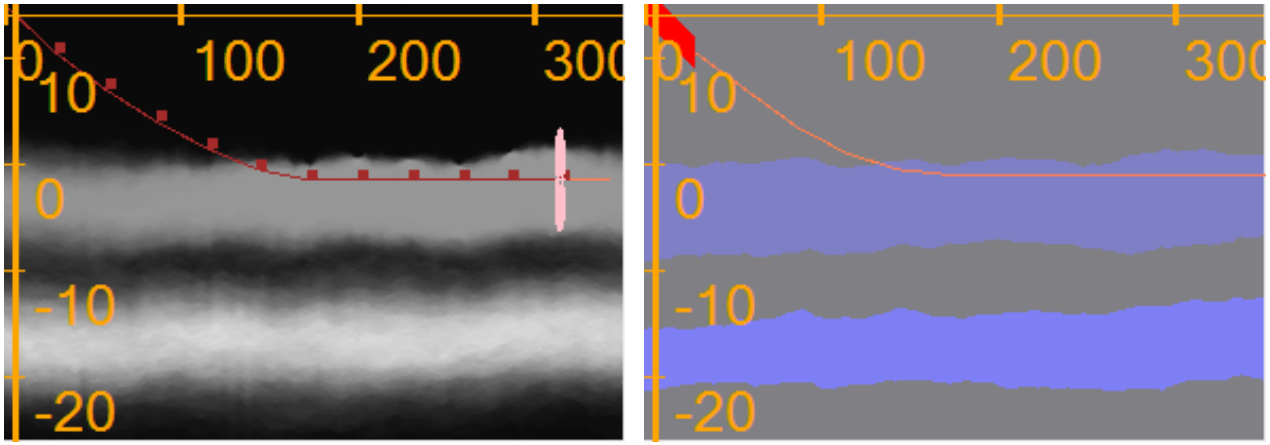
Case A:



- The well matches the perfect trajectory

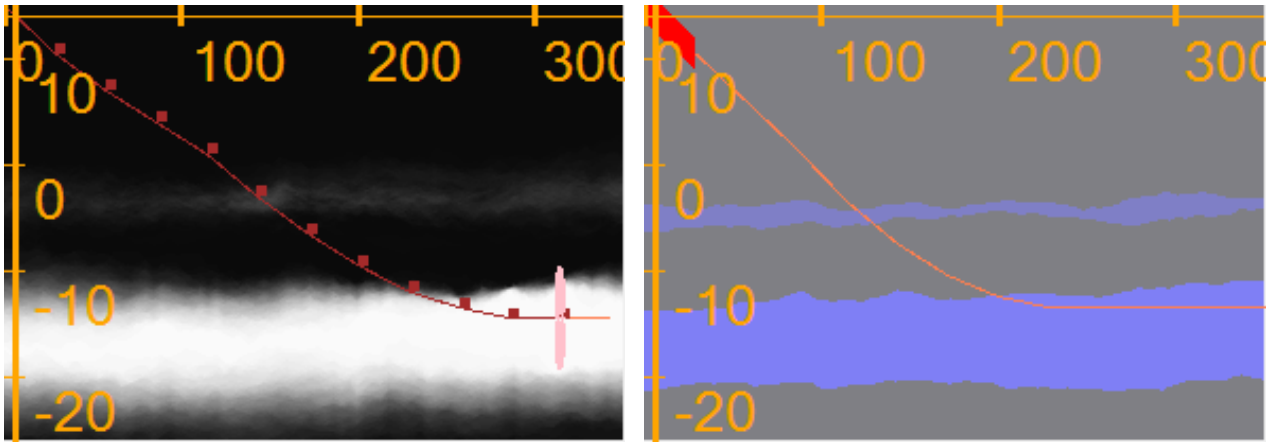
Example 1: Final state

Case A:



- The well matches the perfect trajectory

Case B:



- The well is landed in optimal layer
- The landing is not perfect due to initial uncertainty

Example 2: Interactive DSS interface

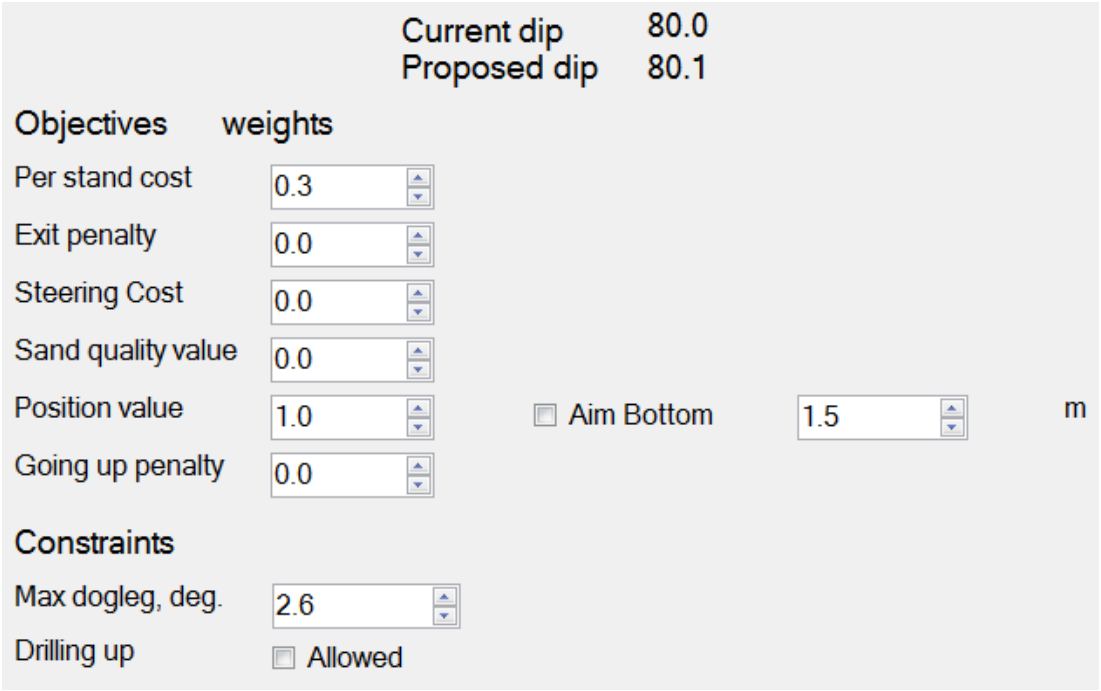


Figure: Elements of GUI

Example 2: Interactive DSS interface

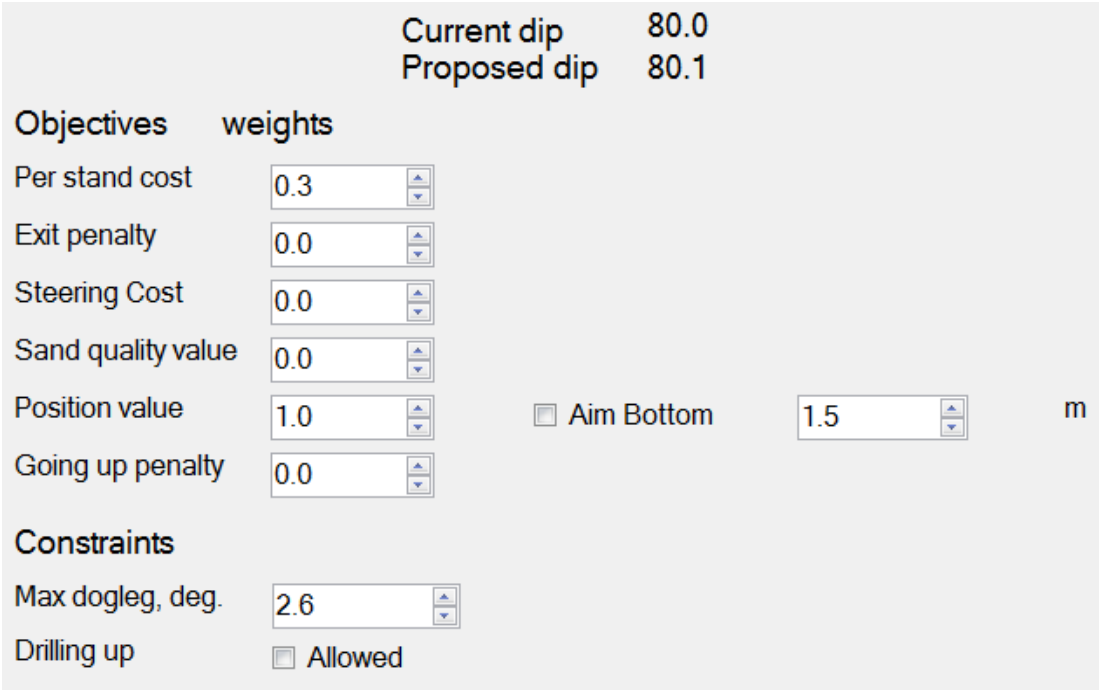


Figure: Elements of GUI

Performance

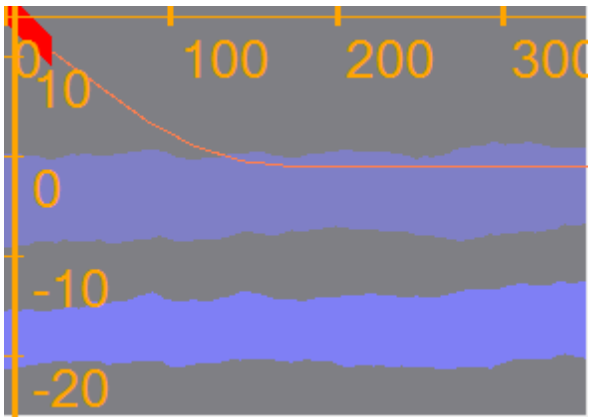
- Incremental model update: 5 seconds
- Recomputation of optimal trajectories: 10 seconds

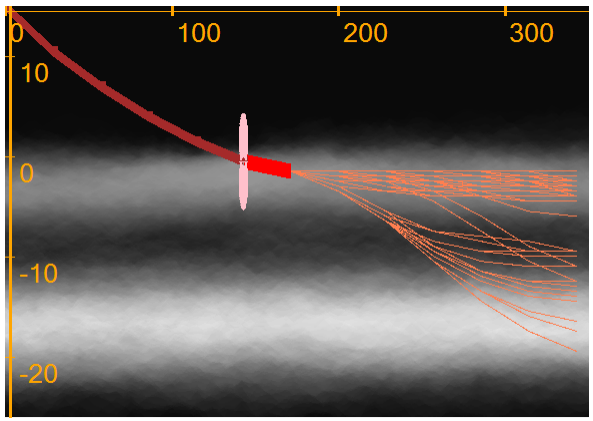
Example 2: Back to step 5

Synthetic truth

Step 5

Case A:





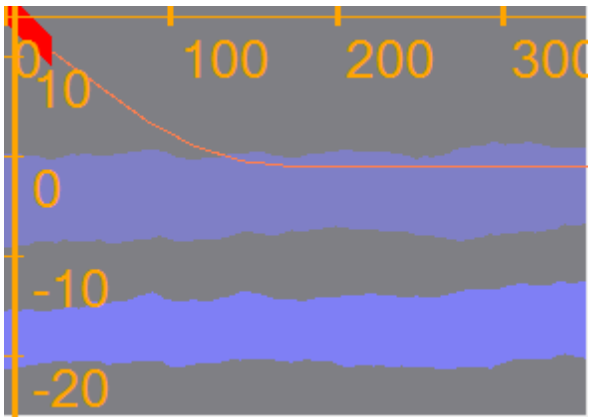
Drill-bit has entered the reservoir

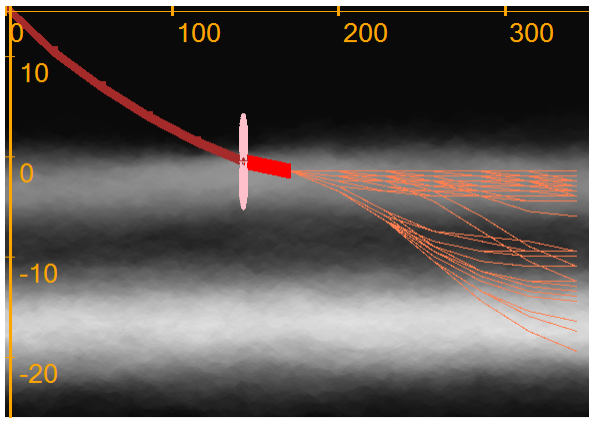
Example 2: Back to step 5

Synthetic truth

Step 5

Case A:





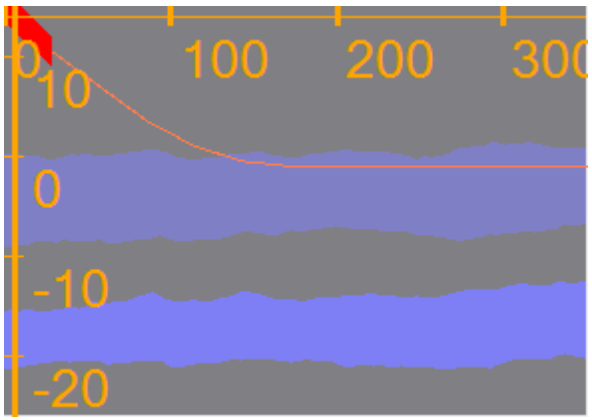
Drill-bit has entered the reservoir

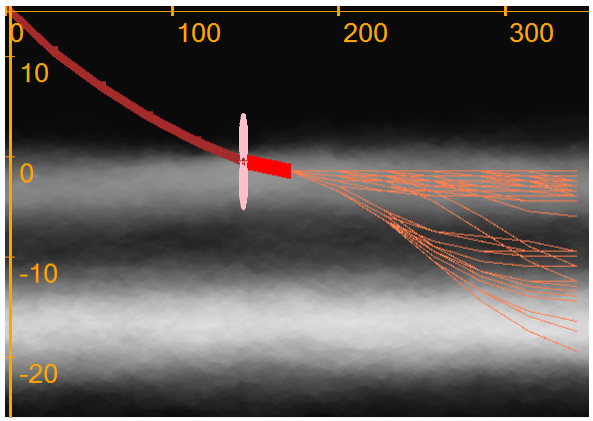
New input: Decision to prioritize "good sand"

Example 2: Back to step 5

Synthetic truth | Step 5

Case A:



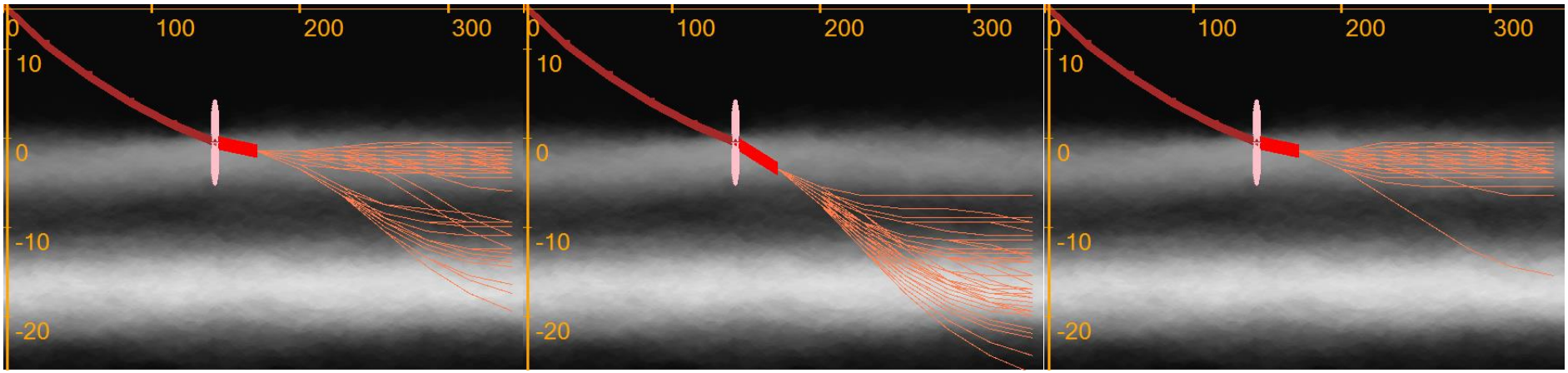


Drill-bit has entered the reservoir

New input: Decision to prioritize "good sand"

- Recomputation of optimal trajectories: 10 seconds
- Preview of outcomes: instant

Example 2: Adjusting objectives due to insights

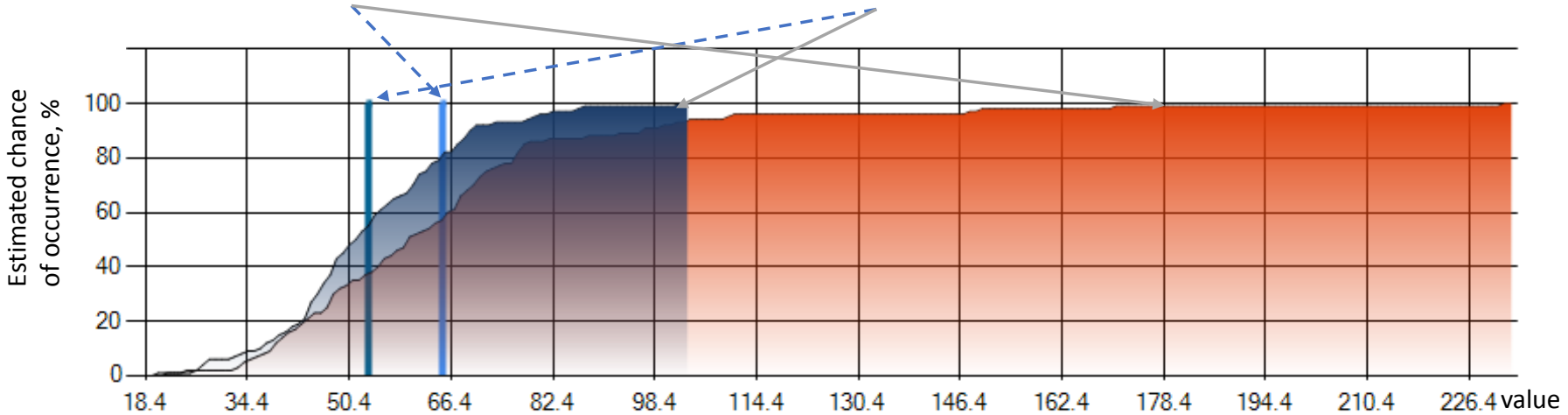


Different choices of weights:

Position: 1.0
Good sand: 0.0
Exit penalty: 0.0

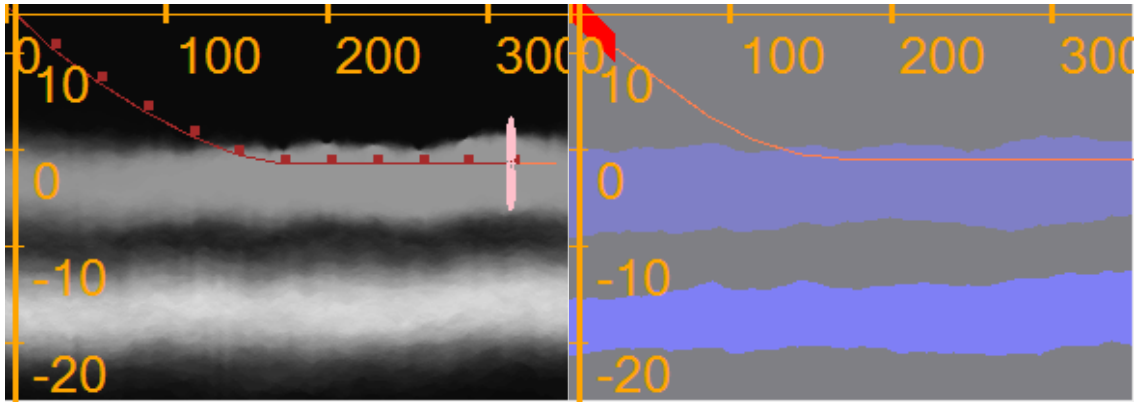
Position: 0.3
Good sand: 0.7
Exit penalty: 0.0

Position: 1.0
Good sand: 0.0
Exit penalty: 1.0

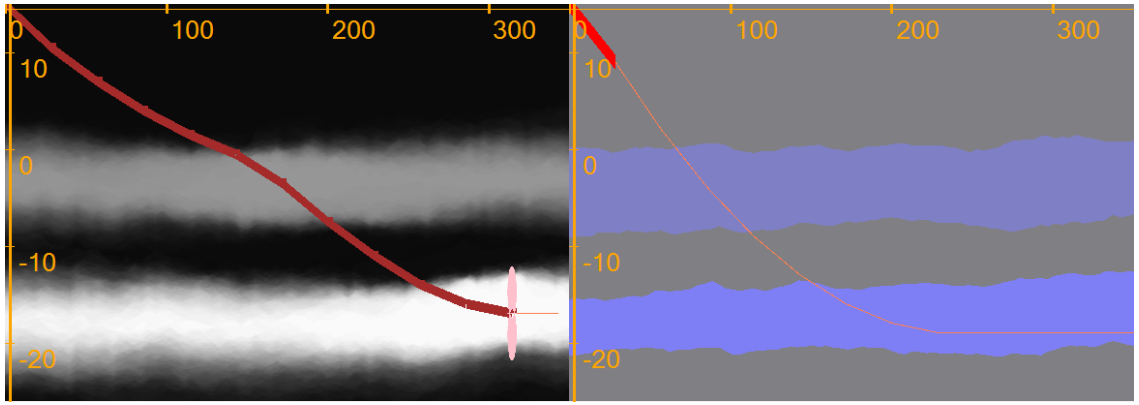


Example 2: Adjusting objectives — Outcome

Case A:

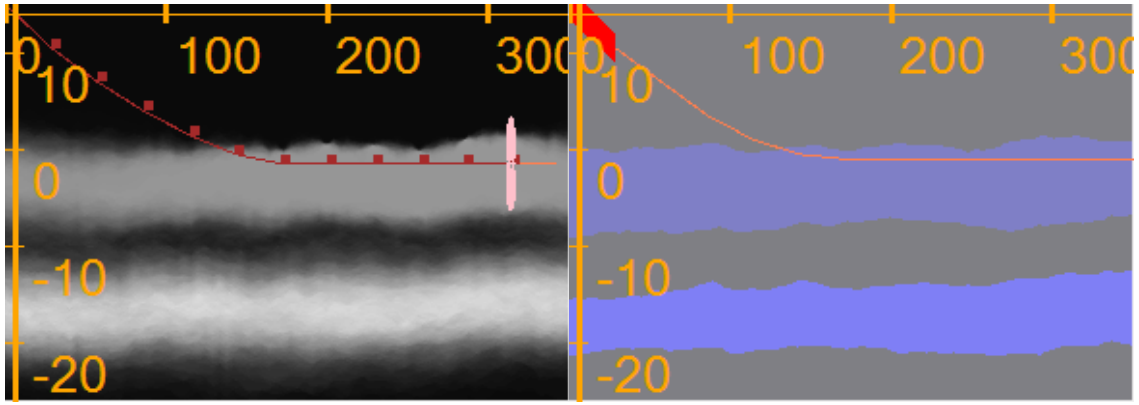


Case A with new metric:

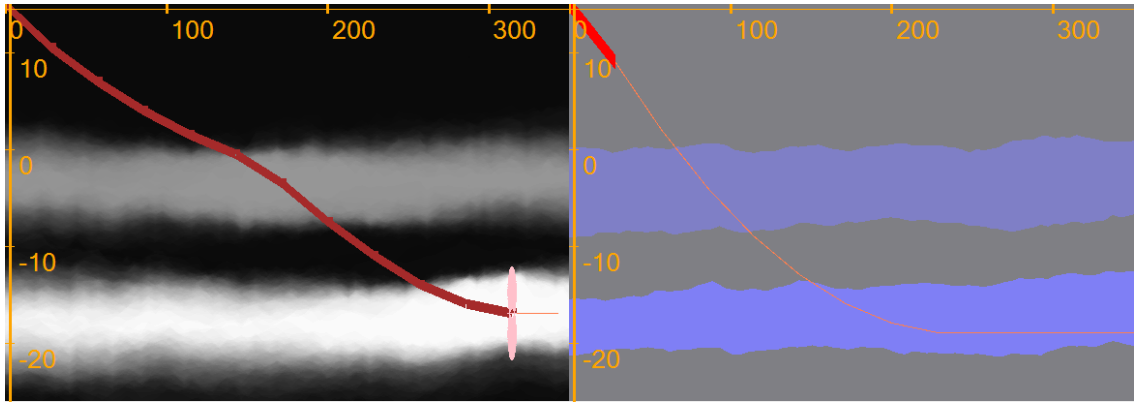


Example 2: Adjusting objectives — Outcome

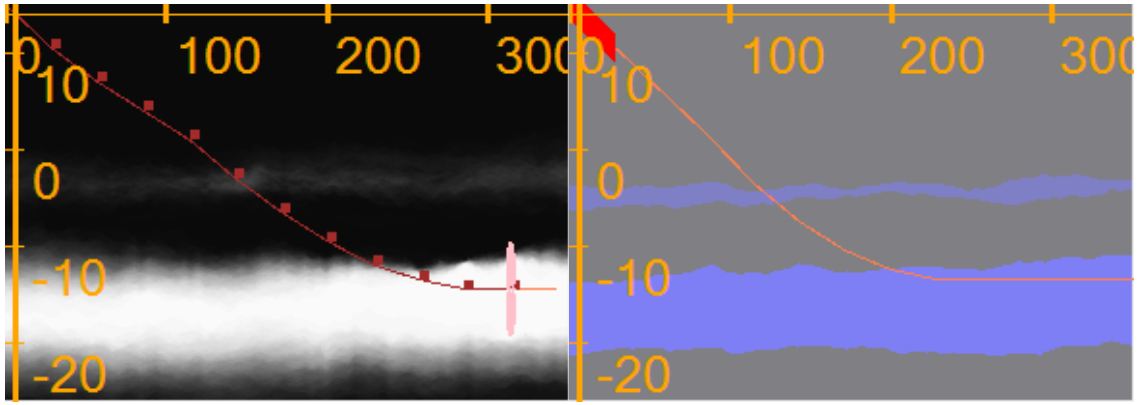
Case A:



Case A with new metric:

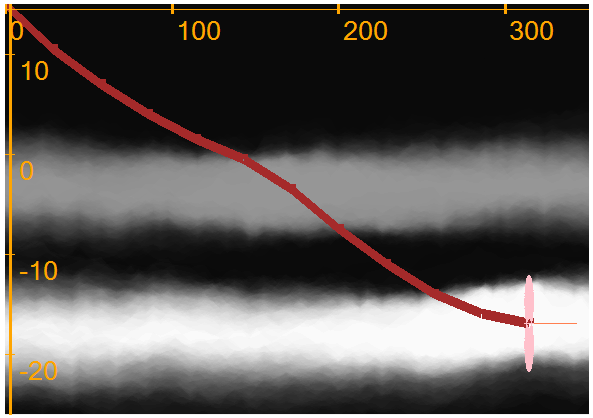


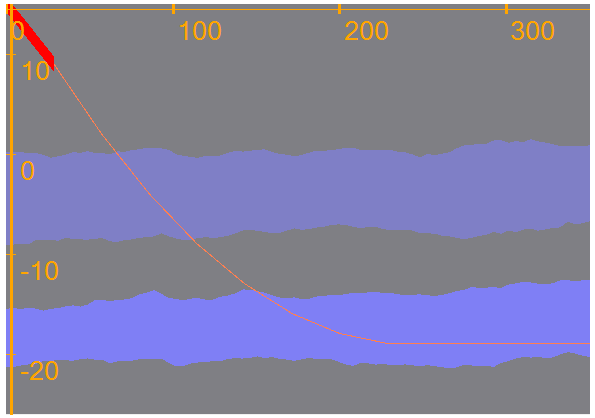
Case B:



Example 2: Adjusting objectives — Summary

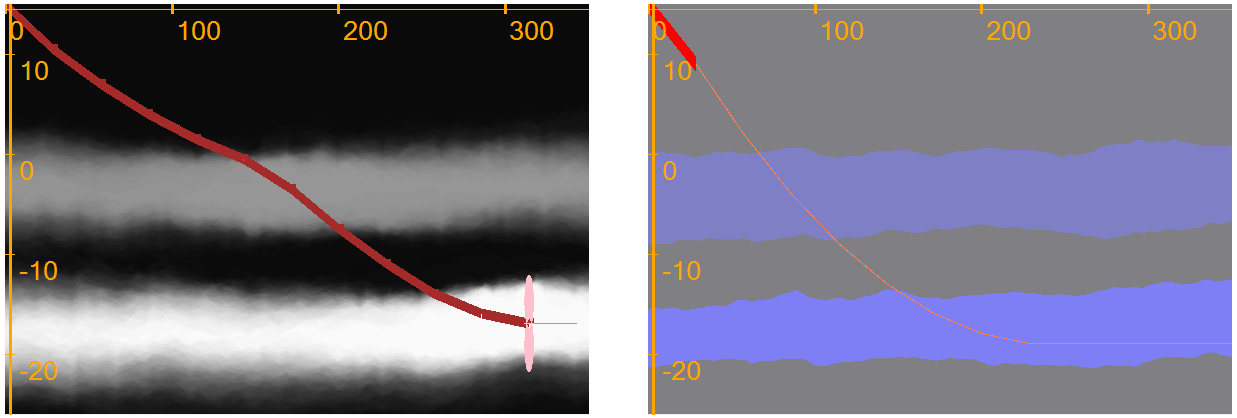
Case A with new metric:





Example 2: Adjusting objectives — Summary

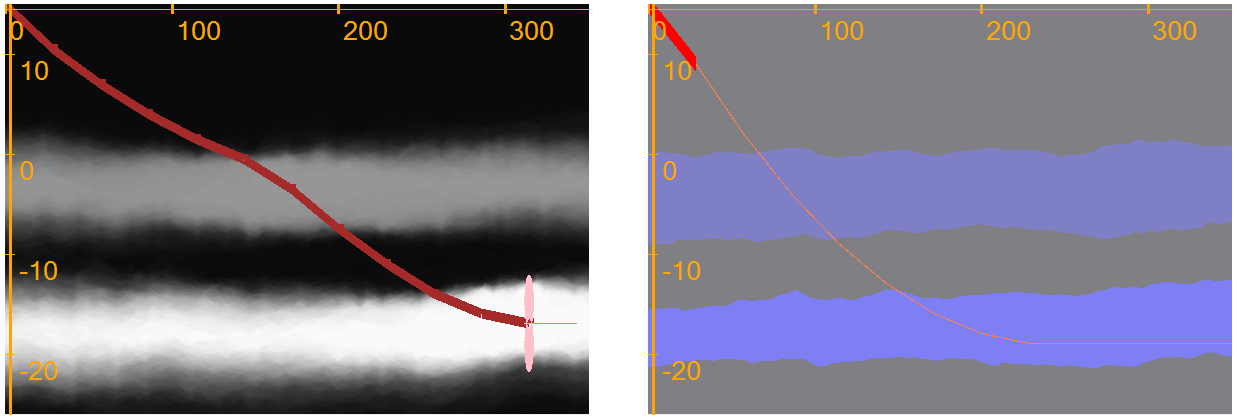
Case A with new metric:



- The well is diverted to 'new optimal' layer following user input

Example 2: Adjusting objectives — Summary

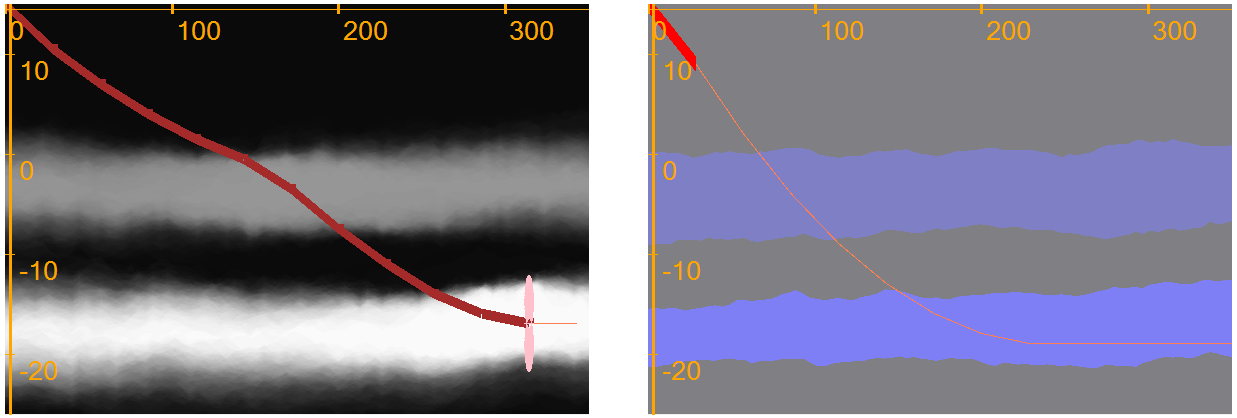
Case A with new metric:



- The well is diverted to 'new optimal' layer following user input
- The new decisions are optimal with respect to new objective

Example 2: Adjusting objectives — Summary

Case A with new metric:



- The well is diverted to 'new optimal' layer following user input
- The new decisions are optimal with respect to new objective
- Providing the correct objective before operation would improve outcomes

Conclusions

Conclusions

We have presented

Conclusions

We have presented

- Ensemble-based update workflow

Conclusions

We have presented

- Ensemble-based update workflow
- Real-time Decision Support System
  - Builds on existing tools
  - Considers full trajectory ahead of drill-bit
  - For each updated realization

Conclusions

We have presented

- Ensemble-based update workflow
- Real-time Decision Support System
  - Builds on existing tools
  - Considers full trajectory ahead of drill-bit
  - For each updated realization

  $\Longrightarrow$ yielding consistently good decisions

Conclusions

We have presented

- Ensemble-based update workflow
- Real-time Decision Support System
  - Builds on existing tools
  - Considers full trajectory ahead of drill-bit
  - For each updated realization

  $\Longrightarrow$ yielding consistently good decisions
- Flexible implementation with intuitive controls and real-time preview of outcomes

Conclusions

We have presented

- Ensemble-based update workflow
- Real-time Decision Support System
  - Builds on existing tools
  - Considers full trajectory ahead of drill-bit
  - For each updated realization

  $\Longrightarrow$ yielding consistently good decisions
- Flexible implementation with intuitive controls and real-time preview of outcomes

[Our paper: Alyaev et.al. (2018). An Interactive Decision Support...]

Thank you!

Acknowledgments

The work was performed as part of the research project **'Geosteering for improved oil recovery'** (NFR-Petromaks2 project no. 268122) supported by the Research Council of Norway, ENI Norge, Statoil and Baker Hughes Norway.

[Our paper: Alyaev et.al. (2018). An Interactive Decision Support...]