Constrained Smoother Path Oscillation issue analysis

Merzlyakov Alexey

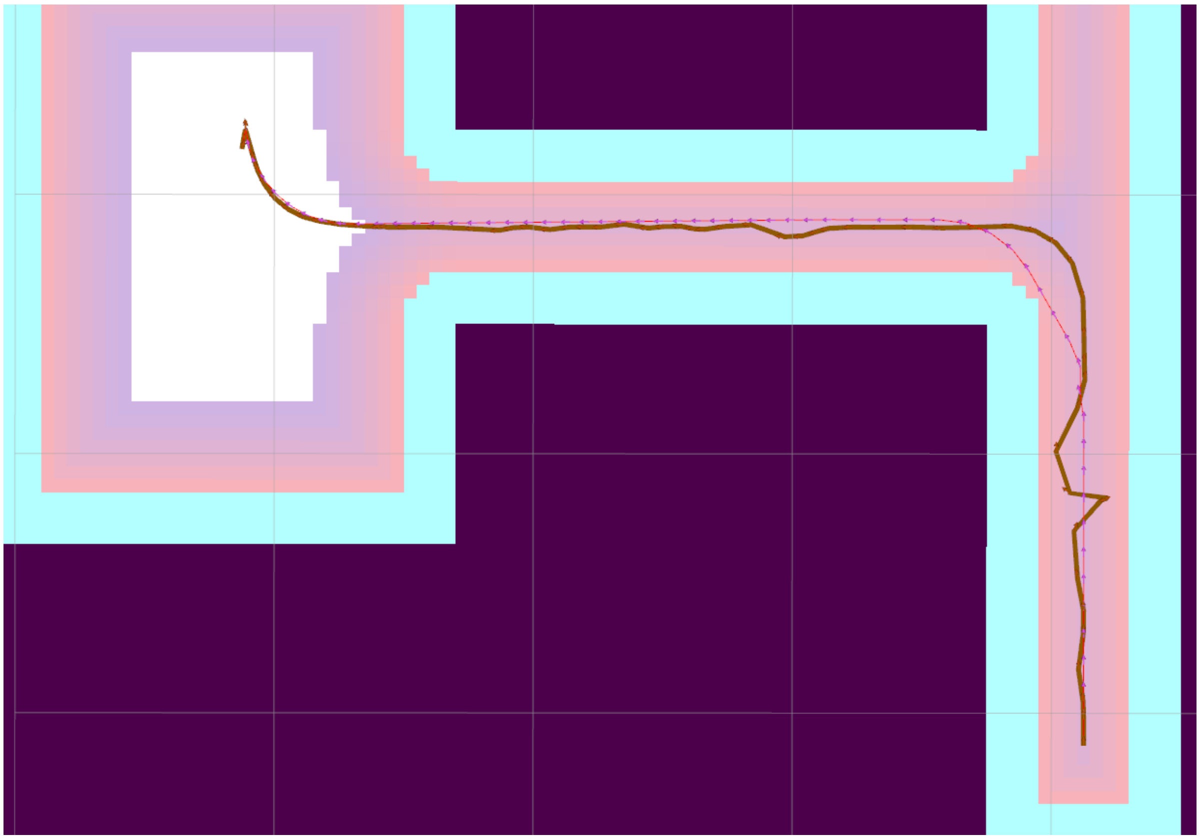
Samsung R&D Institute Russia

Dec 2022

Table of Contents

Background

During the benchmarking of Path Smoothers, it was discovered abnormal behavior of Constrained Smoother. This behavior appeared on some experiments with SmacHybridA* planner and Constrained Smoother, where the path smoothed by Constrained Smoother could have some “wobblings” or “oscillations” like depicted below:



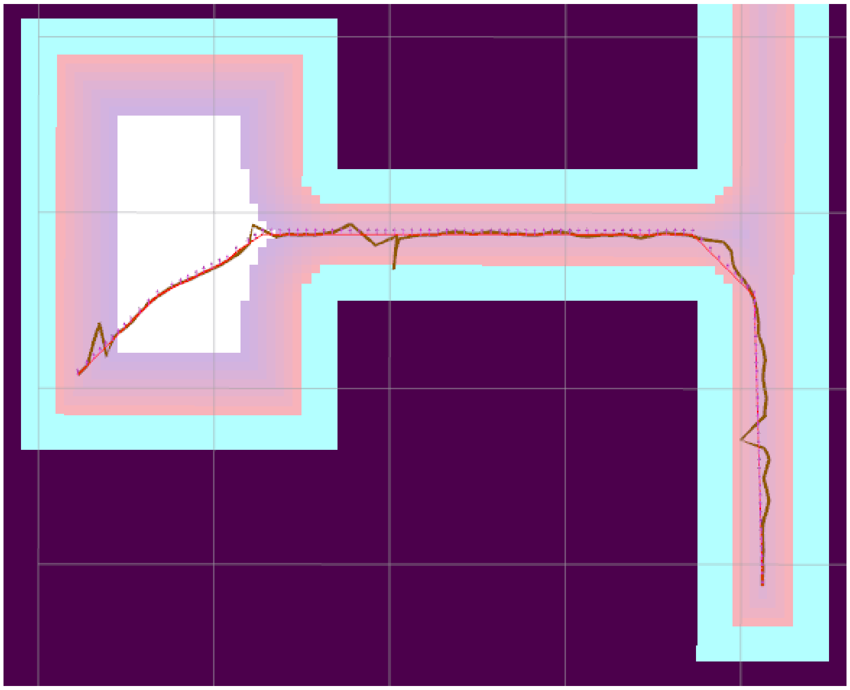
Path oscillation issue. Raw path produced by SmacHybridA planner colored in red, smoothed path by Constrained Smoother colored in brown.*

For this case and all other similar unless otherwise stated:
- SmacHybrid planner was used `REEDS_SHEPP` motion model with `reverse_penalty` set to 1.0 (in order to allow robot to move in equally: backward and forward with the same probability) and other standard parameters.
- Constrained Smoother has also has all standard options set.

This particular path was selected as one of the most complex and thus representative though the experimentation, clearly showing the oscillation issue and to be used in much cases in this analysis.

Path planner effect check

Initially the problem appeared on Smac Hybrid planner. However, changing the path planner to ThetaStar planner (with standard parameters set for this and all examples below, unless otherwise stated) leading to the same effect:



Path oscillation issue on ThetaStar. Raw path produced by ThetaStar planner colored in red, smoothed path by Constrained Smoother colored in brown.

This confirms that the problem is not related to any particular path planner and being a Constrained Smoother issue that requires its analysis. Chapters below are describing this analysis.

Constrained Smoother algorithm

The main algorithm of Constrained Smoother – is a non-linear equation solving problem by using Ceres library finding the local minimum of smoothness function `F`. The function consists from 4 residuals and is being calculated as follows:

$$F = w\_smooth * smoothness^2 + w\_curve * K_{curve}^2 + w\_dist * dist\_deviation^2 + w\_cost * cost^2$$

where:

- `smoothness` – is a length of vector change between next and current path segments, calculated by $|\Delta x_{i+1} - \Delta x_i|$ formula, where
  - Δx_i, Δx_{i+1} - current and next path segments: $\Delta x_i = x_i - x_{i-1}$, $\Delta x_{i+1} = x_{i+1} - x_i$
- K_{curve} – is a curvature coefficient, calculated by $1/R_t - 1/R_{min}$ formula, where:
  - R_t - turning radius in currently smoothed point
  - R_{min} - minimum allowed turning radius
- `dist_deviation` – is a distance between smoothed and original point $|x_i - x\_orig_i|$
- `cost` – is a costmap value approximated for currently smoothed point
- `w_smooth`, `w_curve`, `w_dist` and `w_cost` – are residuals coefficients

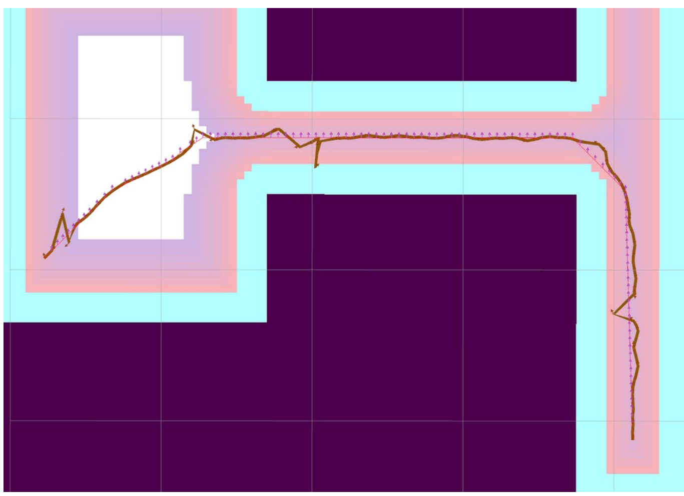
According to the internal algorithms, Ceres solves the problem by iteration algorithm. When the gradient of function `F` change is less than given tolerance, it is considered that the problem is being solved and calculations stop.

The Constrained Smoother is a some way the wrapper under Ceres solving algorithm, so finding the cause of path oscillation problem in Constrained Smoother - is a Ceres problem solver algorithm analysis.

Solver iterations analysis

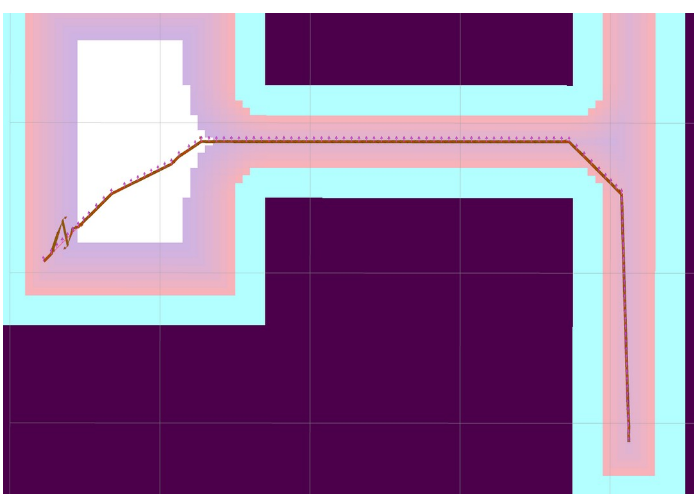
One of the possible pitfall where the Ceres solving algorithm might produce oscillation path effects, could be related with the iterations restriction in main problem solving algorithm. If the oscillations appear on last iterations or wise-versa be restricted by iterations number, this could be the clue for the problem. Let's consider how the smoothed oscillating path is being changed during the Ceres solver algorithm iterations.

For the analysis, ThetaStar planner is being used used with default option set. For Constrained Smoother maximum number of iterations was set to the default 100 value which gave the following path oscillation results:



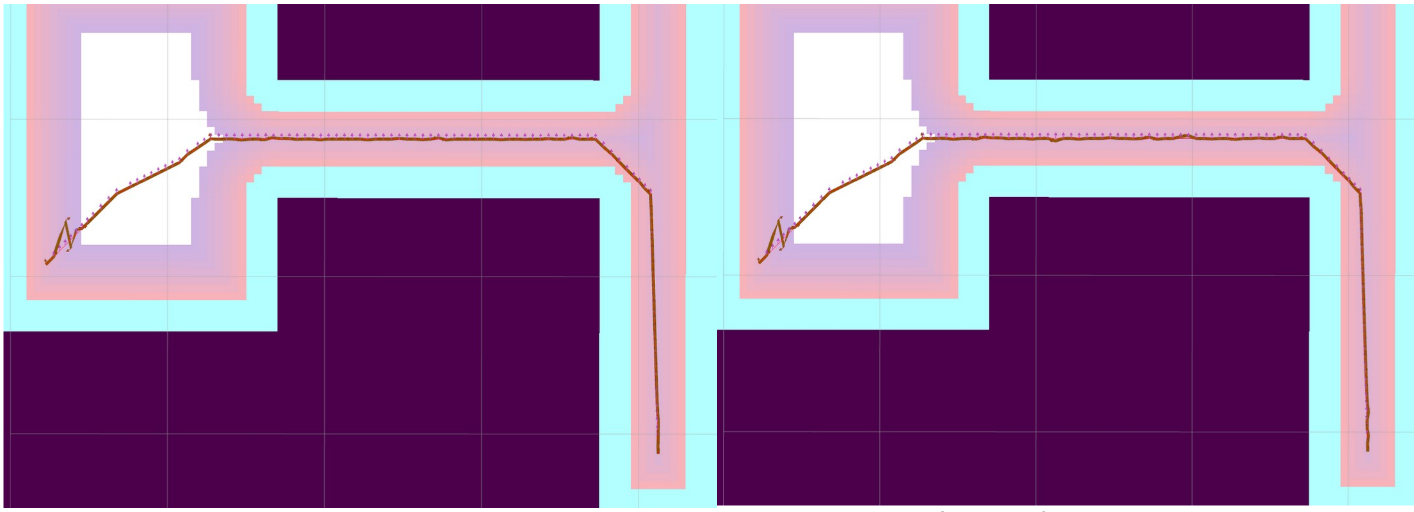
ThetaStar path smoothed by Consrained Smoother, `max_iterations` *is set to 100*

The first peak appear on the 7th iteration of the Ceres main algorithm smoothing:

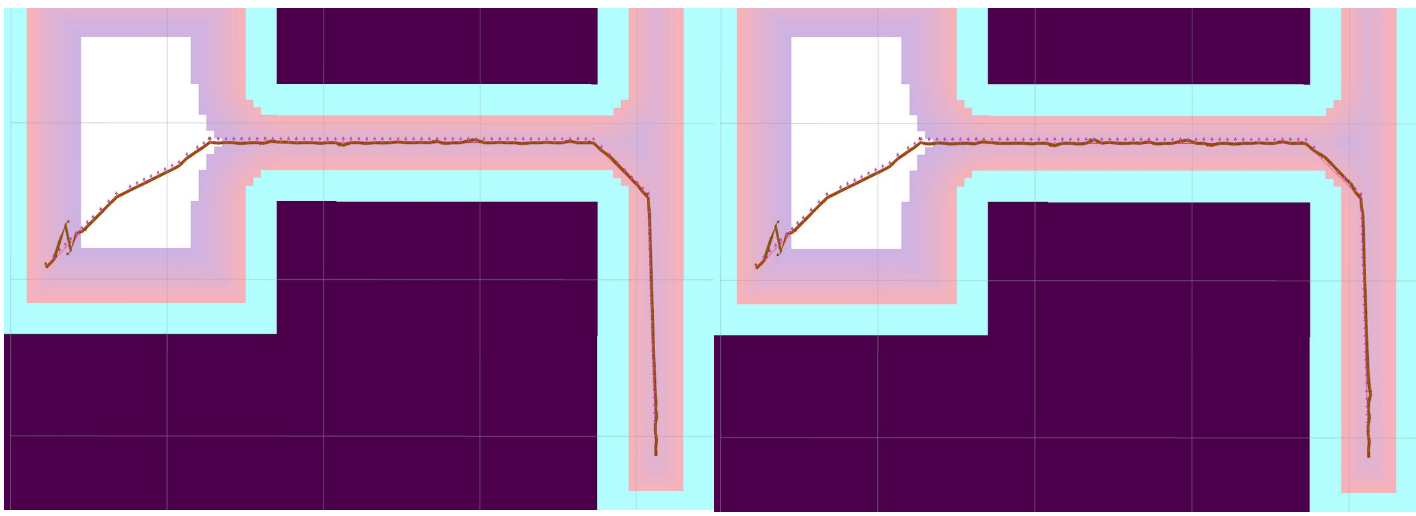


ThetaStar path smoothed by Consrained Smoother, peak appeared on 7th iteration

Next oscillations are started to appear during 55th..59th iterations:

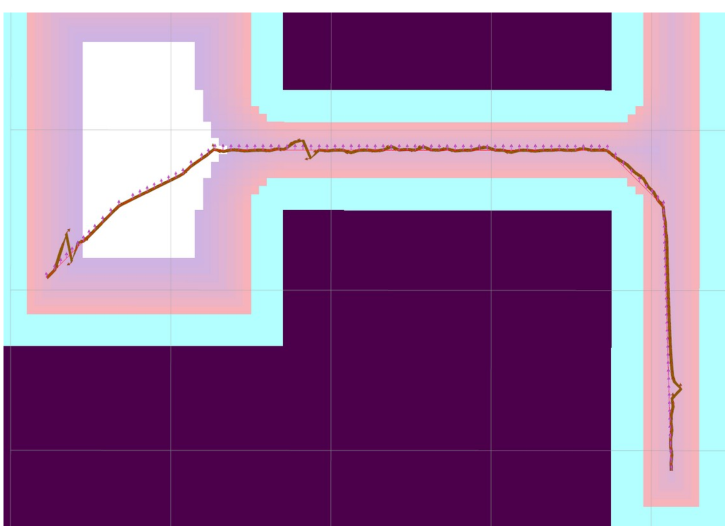


ThetaStar path smoothed by Consrained Smoother, oscillations increased on 55th and 57th iterations



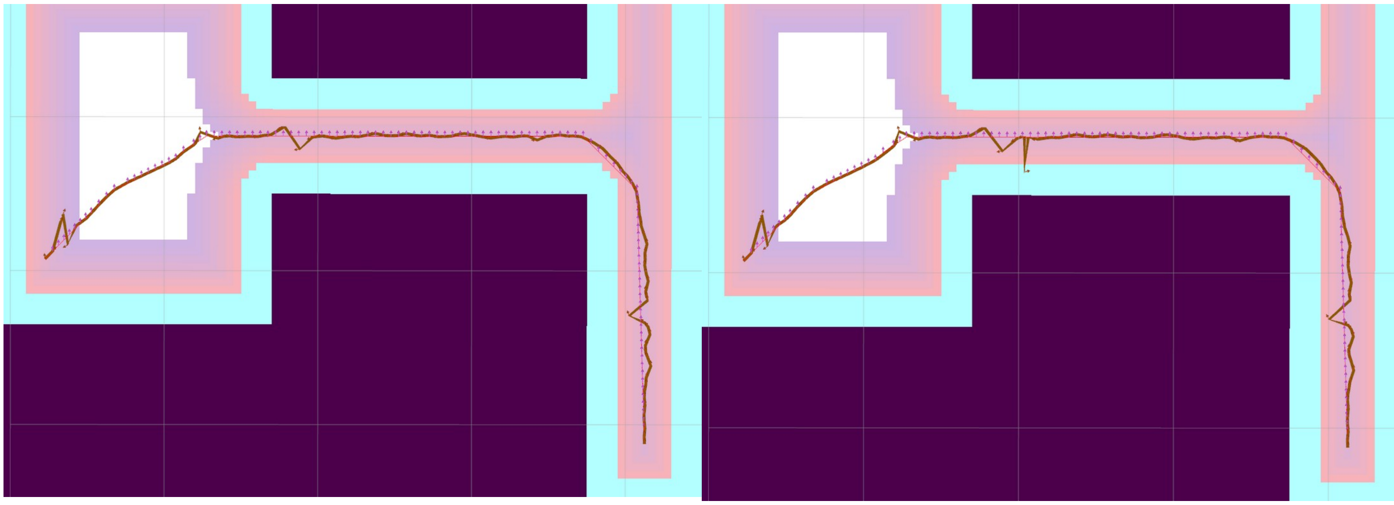
ThetaStar path smoothed by Consrained Smoother, oscillations increased on 58th and 59th iterations

Two peaks (one in the oscillations area, another – in the beginning) are added on the 60th iteration:



ThetaStar path smoothed by Consrained Smoother, next 2 peaks appeared on 60th iteration

Finally, last two peaks in the smoothed path appear on 70th and 71th iterations as follows:



ThetaStar path smoothed by Constrained Smoother, last 2 peaks appeared on 70th and 71th iterations

This experiment shows that oscillations of smoothed path are appearing sequentially through all the cycles increase without any specific reference to the first or last iterations.

Setting the maximum number of iterations for Constrained Smoother to very high value (10000) also gives no effect: path "wobblings" are still appear on the selected testcase.

This mean that path oscillation is not related to any spurious/extreme iteration or top-limiting of max iterations number.

Tolerance effect analysis

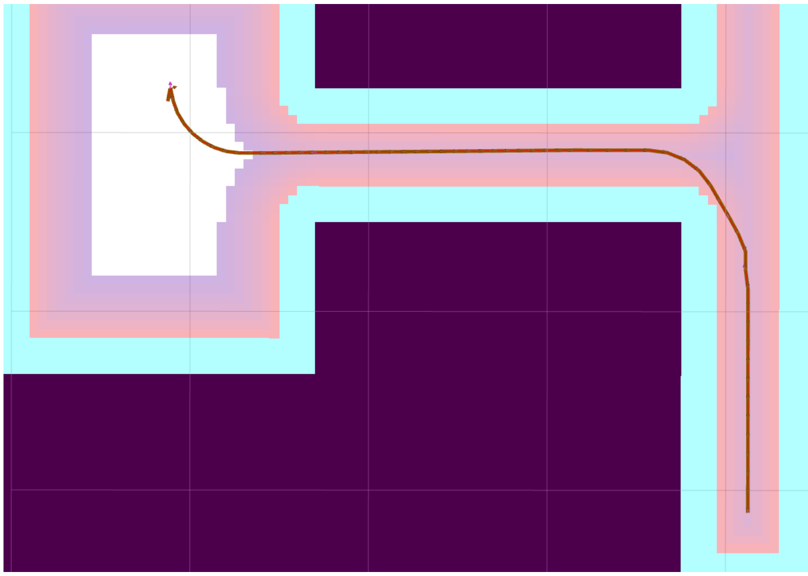
Another possible pitfall might be setting of inappropriate tolerance of Ceres solver function minimum or incorrect change in the gradient of descent of the optimization problem.

To check this assumption, tree parameters were varied:

- `optimizer.gradient_tol`
- `optimizer.fn_tol`
- `optimizer.param_tol`

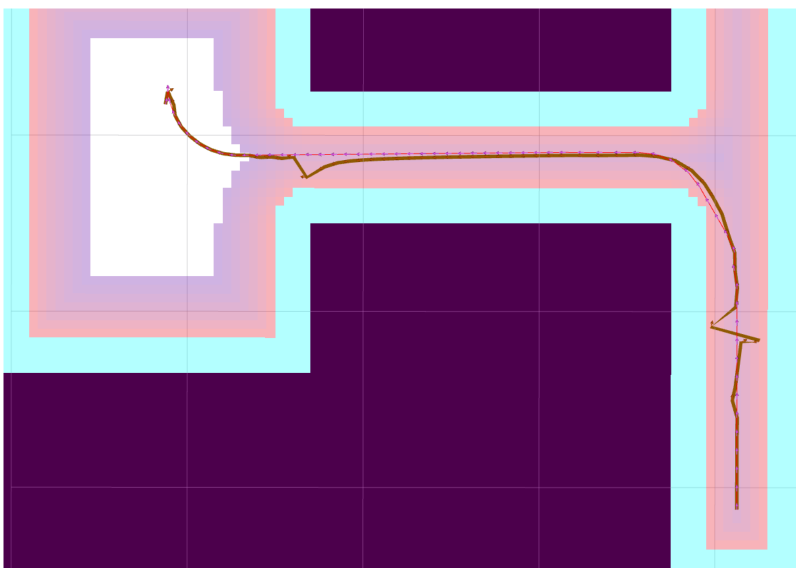
The experiments were proceeded using path, produced by SmacHybridA* planner, tuned for `REEDS_SHEPP` motion model with the `reverse_penalty = 1.0`.

Setting very high values of them giving the smoothed path to match raw path, which is the expected effect – Ceres algorithm won't produce too accurate variations enough for problem solving:



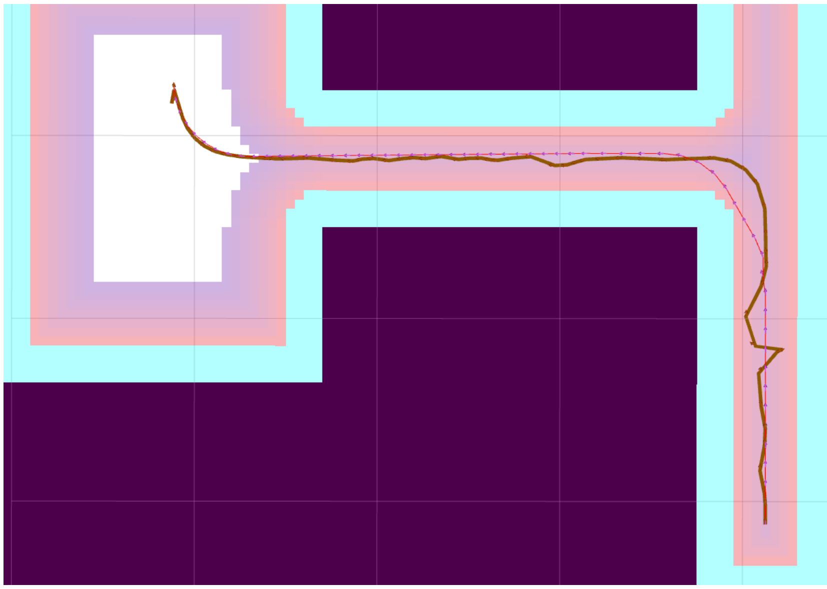
SmacHybrid path smoothed by Constrained Smoother for gradient_tol, fn_tol and param_tol equal to 1e-3

Reduction the tolerances to `gradient_tol` and `param_tol` by 3 orders cause Ceres to produce oscillations:



SmacHybrid path smoothed by Constrained Smoother for gradient_tol, param_tol = 1e-6 and fn_tol = 1e-3

Finally, setting orders of these parameters to the ~half of default values' order causes Constrained Smoother to produce initially observed path oscillations:



SmacHybrid path smoothed by Constrained Smoother for gradient_tol = 1e-5, fn_tol = 1e-4, param_tol = 1e-10

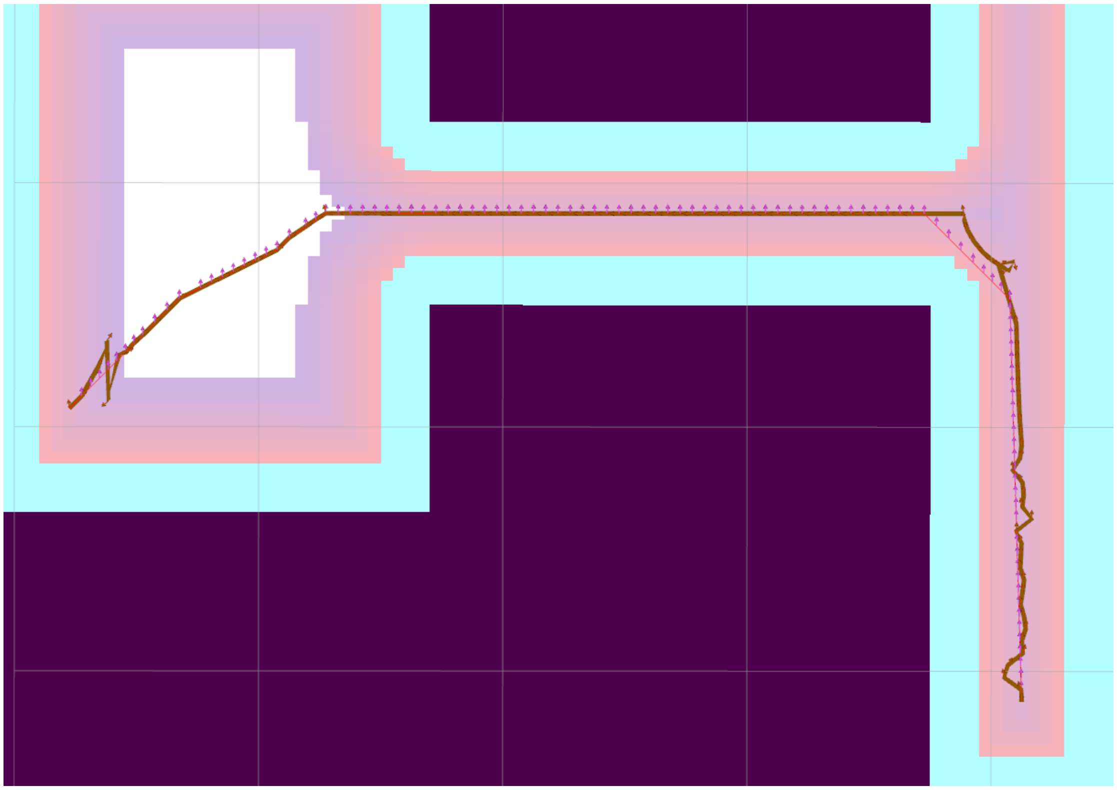
Further reduction of these parameters (down to default values which is `gradient_tol = 1e-10, fn_tol = 1e-7, param_tol = 1e-15`) does not make sense. So, the oscillation effect does not depend of Ceres solver function minimum tolerance or gradient of descent.

Residuals effect analysis

After checking previous assumptions, the effect of what might cause path jitters problem, might be related with Ceres problem function `F` residuals and its coefficients. Let's check one-by-one the effect of contribution to the oscillation problem for each residual in the equation.

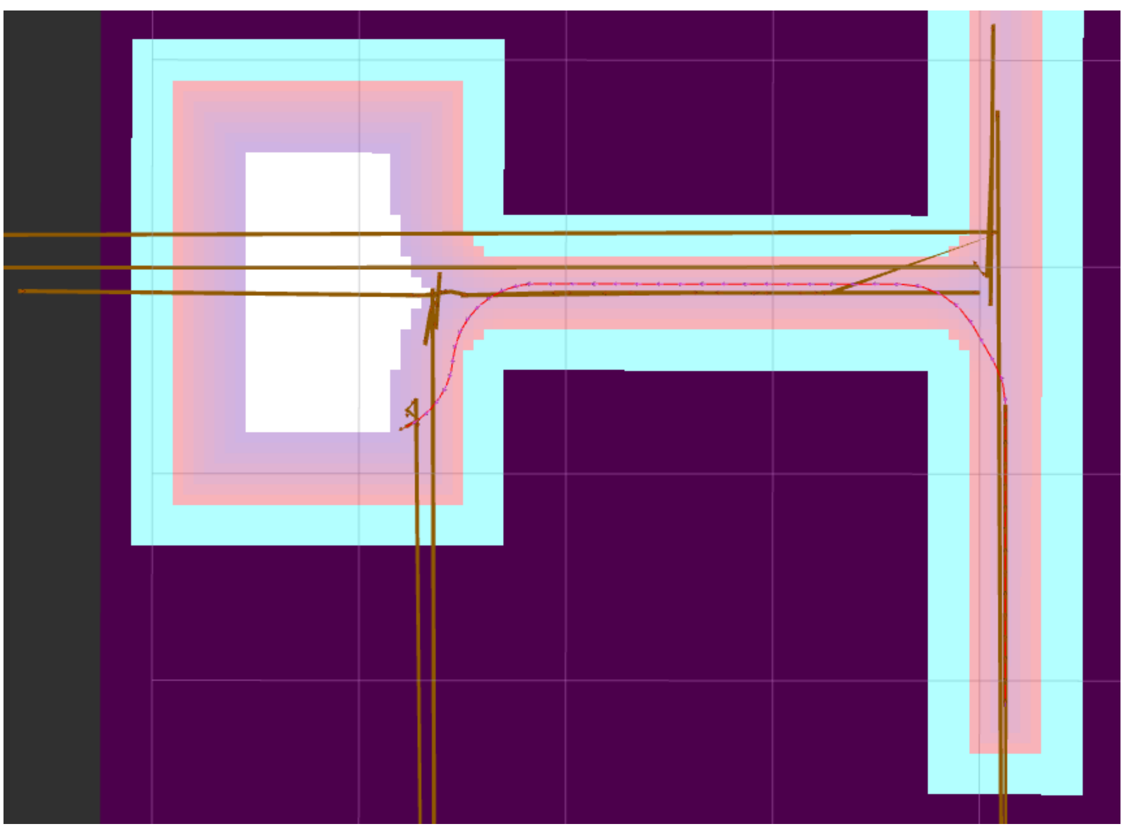
Effect of w_cost

Leaving only the `w_cost` residual from main problem equation (by zeroing rest 3 coefficients) remains the following oscillation effect for the already mentioned in the beginning path produced by ThetaStar planner and smoothed by Constrained Smoother:



Effect of w_cost residual left alone in the equation. Path produced by ThetaStar algorithm.

In other SmacHybridA* path-based example, `w_cost` residual might form more fancy patterns:

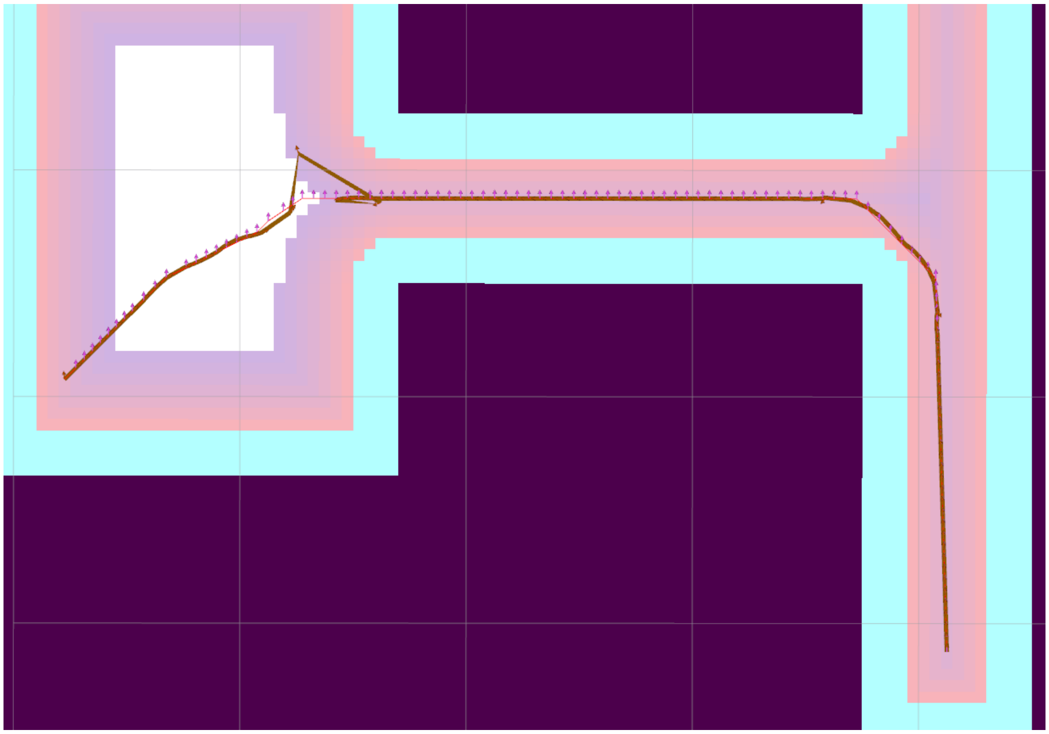


Stronger effect of the w_cost residual left alone in the equation. Path produced by SmacHybridA algorithm.*

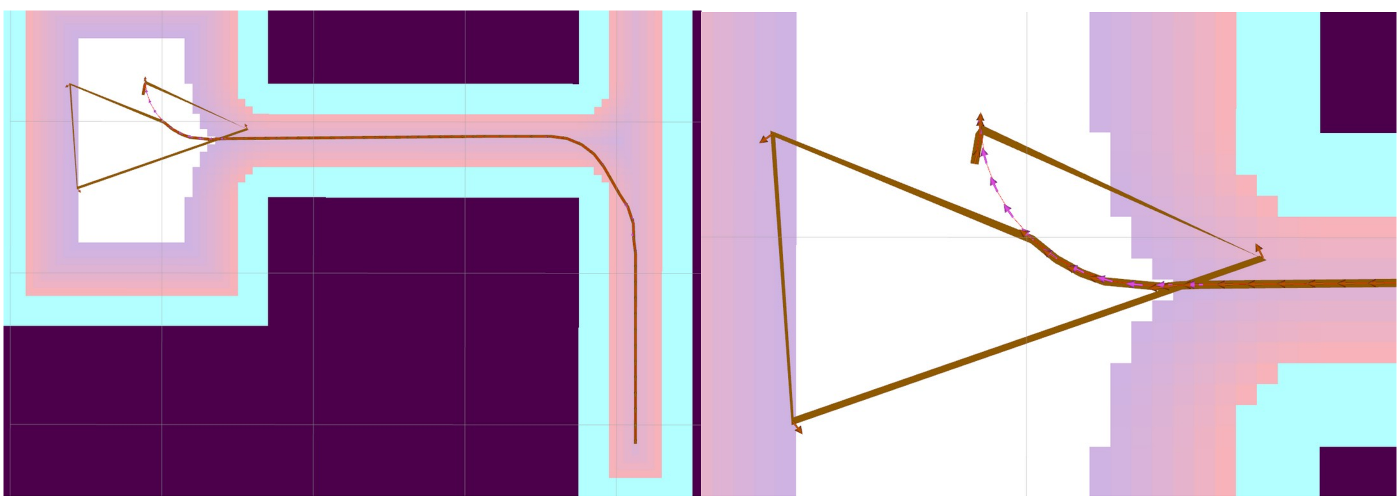
From these observations it can be concluded that `w_cost` component makes a contribution to the final oscillation problem with the selection of other coefficients with relatively small weights to this effect.

Effect of w_curve

The some kind of similar situation is for `w_curve` residual left alone in the main Ceres problem function:



Effect of w_curve residual left alone in the equation. Path produced by ThetaStar planner.



The same effect of w_curve residual left alone for the path produced by SmacHybridA planner.*

Which means that contribution to the oscillation effect might be produced by `w_curve` residual as well.

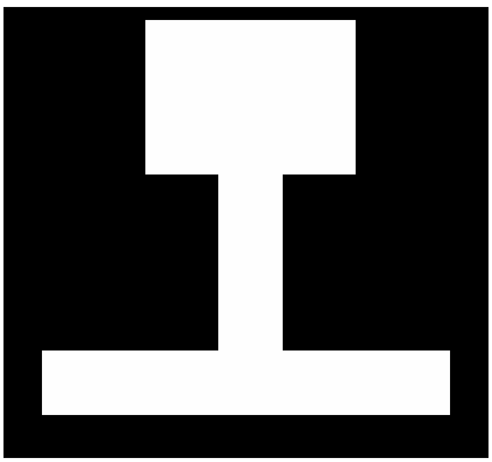
Effect of w_smooth and w_dist

It was observed that both `w_smooth` and `w_dist` are straighten the trajectory, in one case returning/attracting it to the initial raw path (for `w_dist`), in other – making most smoothed trajectory.

Thus, these residuals in the equation are leading to Ceres problem function `F` to produce continuous and more reliable trajectory. The question is in tuning of the `w_cost, w_curve, w_smooth` and `w_dist` coefficients relation, enough for most cases to be suitable.

Comparison of effects probability

In order to estimate the contribution probability for the oscillation effect, each Ceres function `F` residual was benchmarked over 100 randomly generated paths in the Narrow World, which is depicted below:



Narrow World (113x105 cells = 5.65m x 5.25m)

For estimation it was used Path Smoother benchmarking suite placed in the `tools/smoother_benchmarking` directory of Nav2 stack. The benchmarking suite was set to SmacHybridA* path planner (tuned for `REEDS_SHEPP` motion model and `reverse_penalty: 1.0`, as mentioned above) and Constrained Smoother.

During the benchmark each residual coefficient (`w_cost, w_curve w_smooth` and `w_dist`) was one-by-one set to 1.0 while other are being equal to 0.0.

For better representivity, the start pose was fixed for all experiment as was = `(0.9, 0.9, 0.0)` in the beginning of the corridor, while the goal pose (position and orientation) was selected randomly.

All abnormalities in compared cases were detected and summarized in the table below:

Residual	**Abnormal cases**	**Comment**
w_curve	6 out of 100	The trajectories are approximately matching, sometimes with moderate path points scattering
w_cost	100 out of 100	Extreme scattering, like depicted in a “Effect of w_cost” chapter
w_dist	0 out of 100	All trajectories are completely matching

w_smooth	0 out of 100	Smooth and round-shaped trajectories without any binding to the cost

From this table it is seen that abnormalities are mostly caused by `w_cost` residual, although the effect of `w_curve` could not still be ignored.

Code Analysis

Before we jump into tuning of the residual coefficients ratio, we need to verify that these effects are not caused by code bugs or any other issues. Next chapters are trying to shed some light on this.

w_cost code analysis

Since the oscillation effect of `w_cost` appears most frequently, the code related to computation and handling of this residual of Ceres problem function `F`, should be checked.

1. Costmap round check

During the experimentation and debugging, it was discovered that costmap used in a Ceres problem solver is being interpolated by using `ceres::BiCubicInterpolator` approach. This method interpolates the value of costmaps between grids, but in some points between them may cause an inaccuracies, e.g. by producing too large or even negative costs.

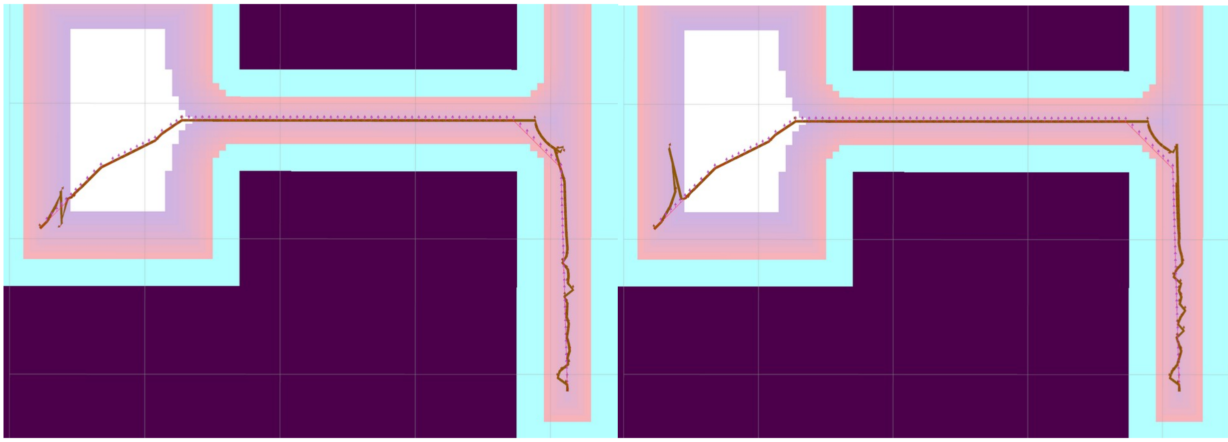
The first idea was to round the values produced by `BiCubicInterpolator` to actual costmap ones by taking floor values in cell's corners where costmaps are taking their actual integer values.

However, after evaluating rounding approach, the path smoothed by Constrained Smoother did not produce any other trajectory rather than original raw path from path planner. This effect could be explained by that Ceres during its iterations requires continuous variation of parameters, including costmap, and making costmap function to be grained will cause such effects.

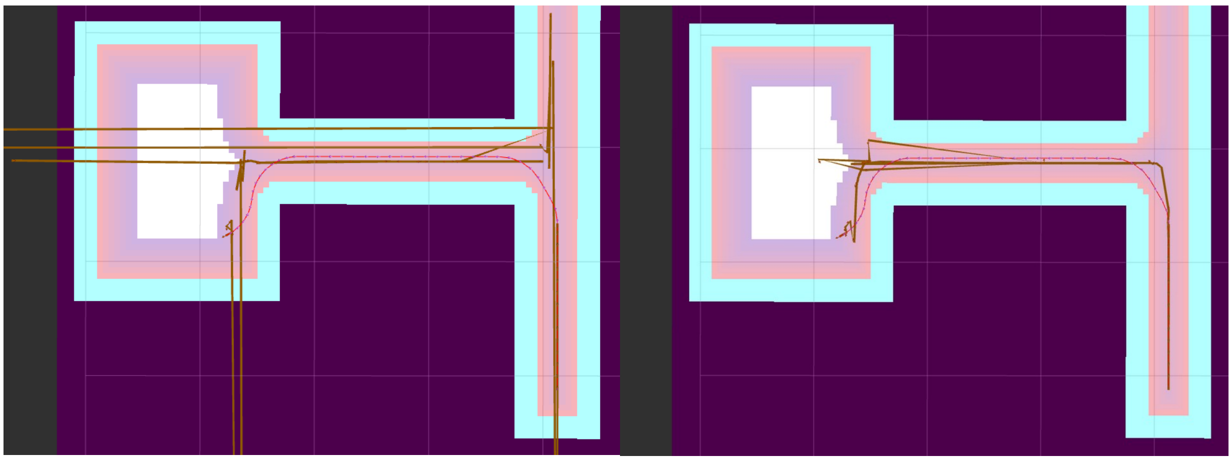
2. Negative cost and out of boundary analysis

Another possible problem could hide in the negative costs (as was mentioned above) and absence of costmap boundaries check during the calculation of `w_cost` residual. The estimation approach was to add negative values of costmaps and costmap boundaries high-cost penalties into the `w_cost addCostResidual()` calculation function.

The path smoothed with only `w_cost` residual enabled and experimental patch applied, produces promising results for ThetaStar and SmacHybridA* planners:



ThetaStar path smoothed with w_cost residual without (left) and with (right) experimental patch applied.



SmacHybridA path smoothed with w_cost residual without (left) and with (right) experimental patch applied.*

However, moving a little forward, if set `w_smooth = 2M` with other standard coefficients set, the approach did a slightly negative effect of less optimal paths discovered on many tests from "Comparison of effects probability" chapter.

w_curve code analysis

Another possible point of problem in poor performance of `w_curve` problem function `F` residual, could be an issue in curvature calculation `addCurvatureResidual()` function. As written in "Constrained Smoother algorithm" chapter, the residual is being calculated as:

$K_{curve} = 1/R_t - 1/R_{min}$ equation,

where R_t and R_{min} are actual and minimum allowed turning radiuses.

In the curvature residual calculation function K_{curve} is being compared with positive neighboring of zero, but only from positive side:

```
if (K_curve <= +ε) { do not add the residual }
```

On the first sight it might seem to be a possible pitfall for the `addCurvatureResidual()` function operating: if K_{curve} will take high negative values, far from 0.0 by module, the curvature residual calculation function still won't produce any results. However, this behavior is intentional: K_{curve} will take negative values if R_t is less than R_{min}, which never should be appear by function design. This check is actually designed to be `if (K_curve <= 0) { do not add the residual }`, but $+\varepsilon$ is added to avoid additional floating-point error caused by previous radius calculations.

So, this part of code seems to have no problem.

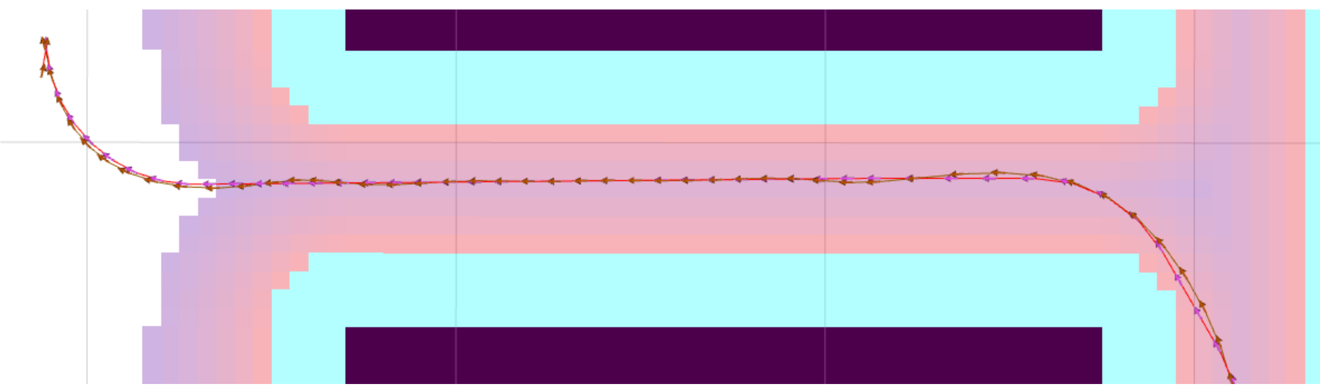
Parameters tune

As of chapters above, the residuals could be divided into the negative- and positive-effect producing parts of the equation. Negative effect-producing parts are **`w_cost`** and **`w_curve`** while positive effect producing are: **`w_smooth`** and **`w_dist`**. Since the **`w_cost`** and **`w_curve`** parts of the Ceres problem function `F` have no knowledge about path continuity and integrity, they should be used along with **`w_smooth`** and **`w_dist`** to produce optimal smoothed paths. Only if both groups of residuals are working together, we could obtain the necessary performance of the Constrained Smoother algorithm.

Since **`w_dist`** by its formula tents to return the path to its initial unsmoothed way, there is no sense to use it in the production. So, let's focus on the tune of **`w_smooth`** opposing to the **`w_curve`** and **`w_cost`** coefficients in next sub-chapters below.

w_smooth vs w_curve

The effect of [initial path], produced by SmacHybridA* planner and smoothed by Constrained Smoother with only enabled `w_smooth` and `w_curve` residuals, is being experimented on fixed `w_curve = 1.0` and varied `w_smooth`. The starting point for `w_smooth` was selected as `w_smooth = 10,000` where notable oscillations are still observed:



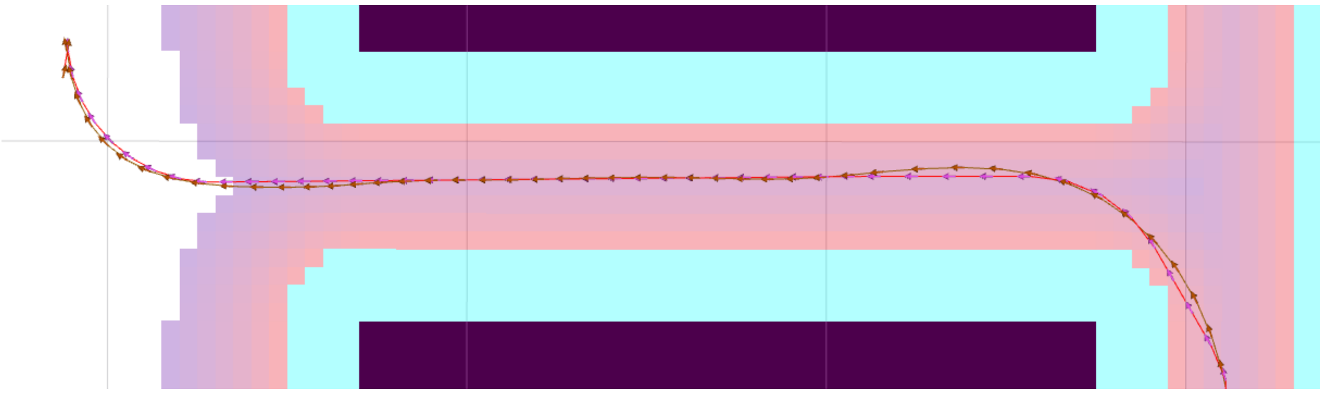
Oscillations for w_smooth = 10K, w_curve = 1 (visible at the left and right sides near the path turning). Path produced by SmacHybridA path planner.*

Increasing the `w_smooth` coefficient on this example is giving the result as follows in the table below:

w_smooth (w_curve = 1.0)	**Quality**
10,000	Notable oscillations
20,000	Notable oscillations
30,000	Suitably good

50,000	Good
100,000	Good
300,000	Excellent

It is seen, that starting from `w_smooth = 30K`, the oscillations are almost disappear on this case:



No oscillations for w_smooth = 30K, w_curve = 1. Path produced by SmacHybridA path planner.*

Further increase of the `w_smooth` in this experiment doesn't make sense: path smoothed by Constrained Smoother in this case will remain smooth and without oscillations.

Factoring w_smooth for standard parameters

As follows in the [sub-chapter above](), the optimal set for `w_curve = 1.0` is to select at least `w_smooth = 30K`.

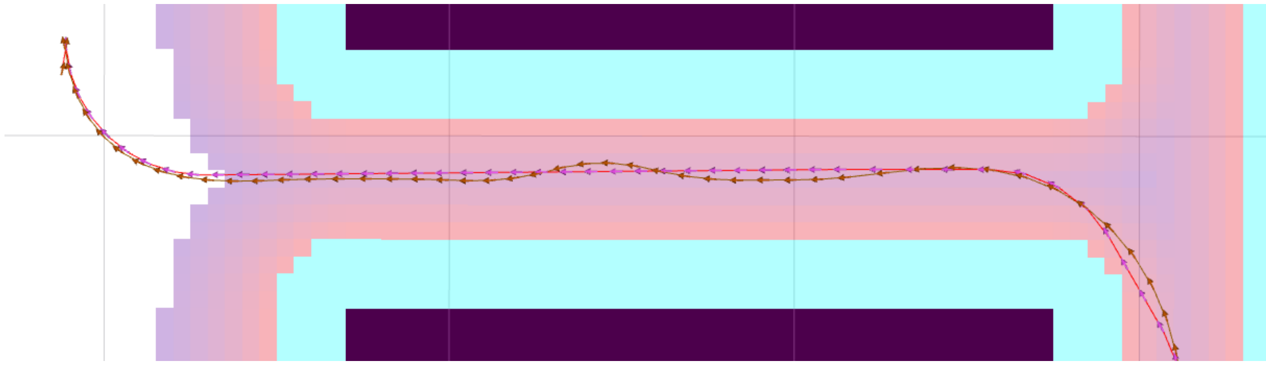
If set `w_curve = 30.0`, as set in the production, `w_smooth` should be multiplied by `x30` and will be equal to `w_smooth = 900K ~= 1M`

Starting from this point, it is worth to carry out next experiments by tuning `w_cost` residual coefficient in the next sub-chapter.

w_smooth vs w_cost and w_curve

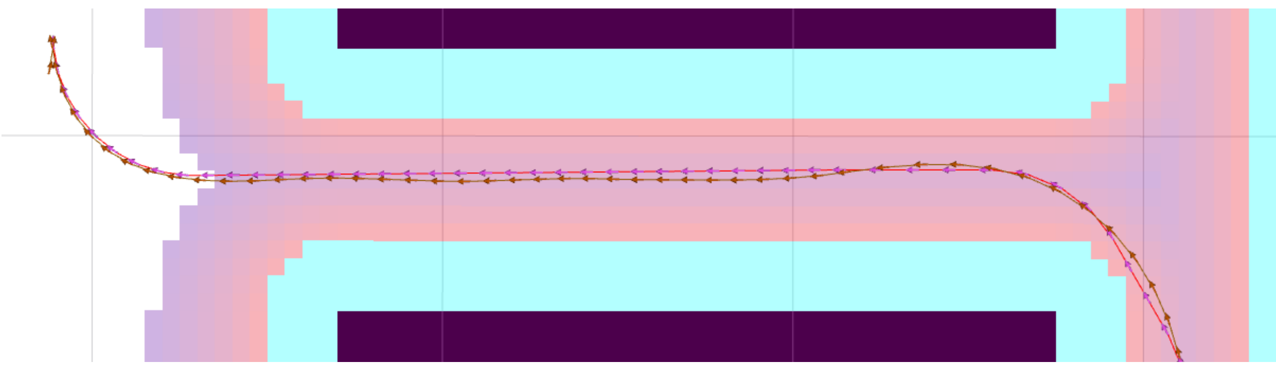
We are adding the `w_cost` factor. In the same experiment, we set `w_cost = 0.015, w_curve = 30.0` as it made in the production, and will vary the `w_smooth` coefficient. Since the probability of oscillation effects cased by `w_cost` and `w_curve` factors [outweighs the first](), the ratio between `w_curve` and `w_cost` is more than justified.

As it was previously discovered, starting point is: `w_smooth = 1M`, which shows notable peak in the middle of the path produced by SmacHybridA* planner and smoothed by Constrained Smoother:



Oscillations for w_smooth = 1M, w_cost = 0.015, w_curve = 30 (peak is visible at the middle of the path)

However, the increase of w_smooth in x2 times improves much the situation:



Almost no oscillations for w_smooth = 2M, w_cost = 0.015, w_curve = 30

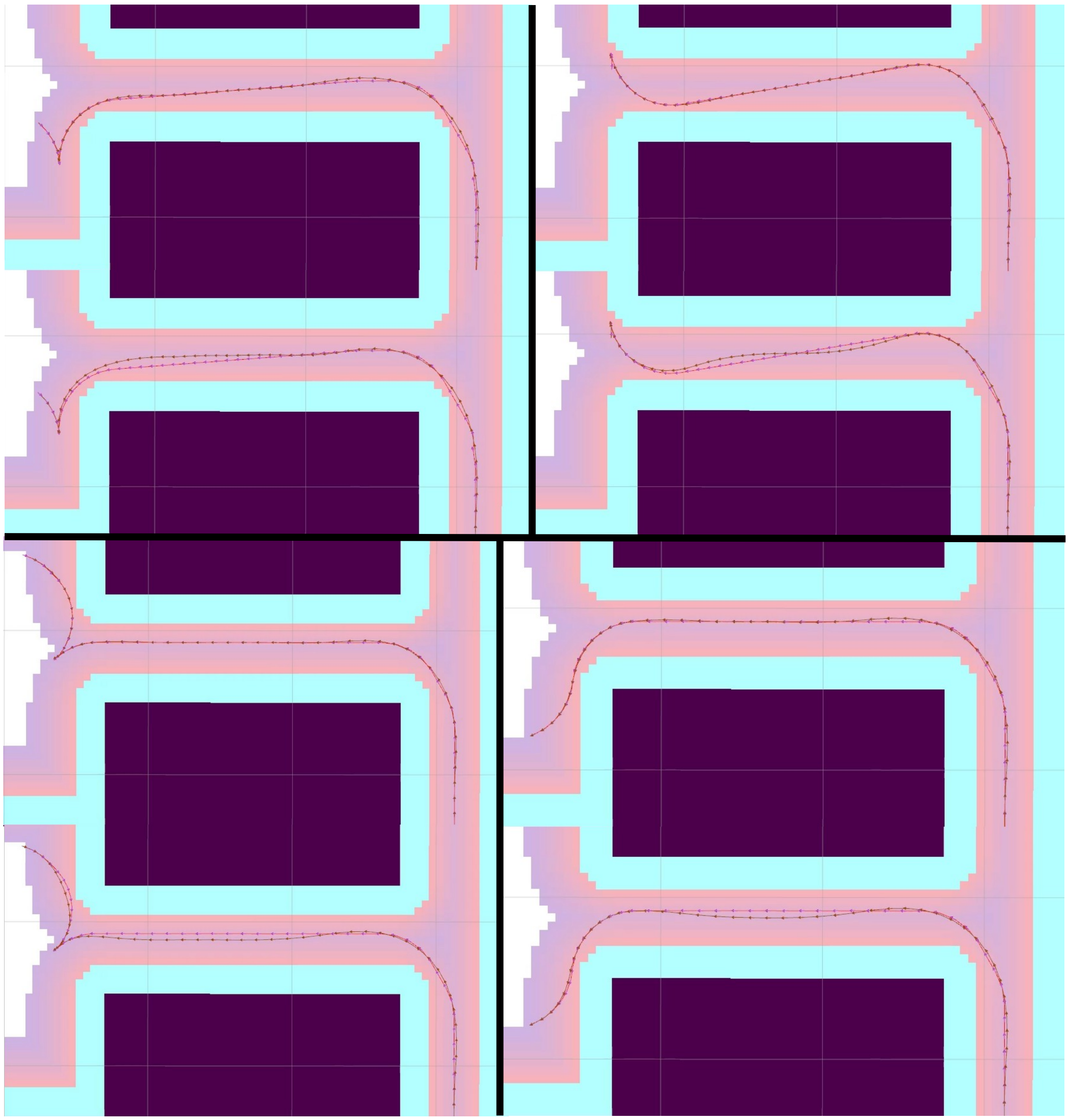
Further increase of the `w_smooth` on this example does not result any significant changes, as represented in the table below:

w_smooth (w_cost = 0.015, w_curve = 30.0)	**Quality**
1M	Notable oscillations
2M	Suitably good
2.5M	Suitably good
3M	Suitably good

So, the minimal suitable value of `w_smooth` where the selected complex case is showing the satisfying results was chosen as `w_smooth = 2M` candidate for the standard options set.

w_smooth relative strength check

In previous chapter it was selected the minimal suitable value of `w_smooth` relatively to other default production residual coefficients. It is turned out that in this optimal ratio, `w_smooth` is 9 order of magnitude greater than `w_cost` and 6 order of magnitude greater than `w_curve`. Such high relation in favor of `w_smooth` can negate any other effects of high costs avoiding or entering into a curve. At this step we are confirming that these effects are still having place, even for such high coefficients ratio. We are using the same as in previous chapter setup. During the experimentation, Constrained Smoother tuned only by `w_smooth` Ceres equation residual, produces the trajectory as smooth as possible (most often following the raw path produced by SmacHybridA* planner), but having no respect to costs. However, the path produced with `w_smooth = 2M` tuned with other parameters set to their default production values, shows the optimal smoothed path with the places having better costs, as follows at the illustrations below:



4 experiments showing the respect to costs and curves with suggested residual coefficients ratio. In each experiment, upper images are having only w_smooth part in the Ceres problem equation, while bottom images are having w_smooth=2M with other w_cost 0.015, w_curve=30.0 parameters set. Raw paths produced by SmacHybridA planner are depicted in red, smoothed paths – in brown.*

So, having `w_smooth` 9 order of magnitude greater than `w_cost` and 6 order of magnitude greater than `w_curve` relation showing that Constrained Smoother still will produce the path in respect to low-costs areas and curves.

Random paths verification

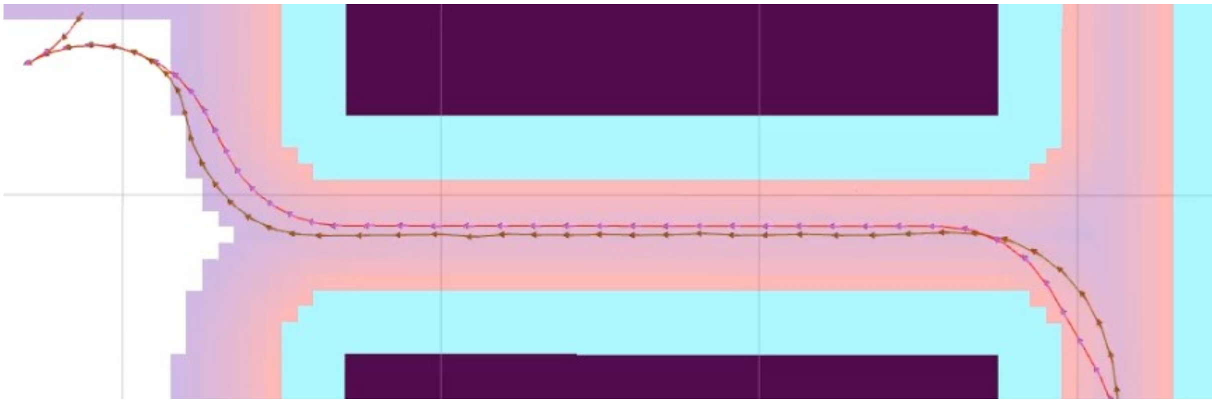
In previous chapters option set was estimated for Constrained Smoother on one of the most complex example. However to represent the reliability of this choice, we need to verify the minimum of possible values and viability of selected **w_smooth** value in comparison to the rest **w_cost** and **w_curve** other randomly generated paths.

Narrow World

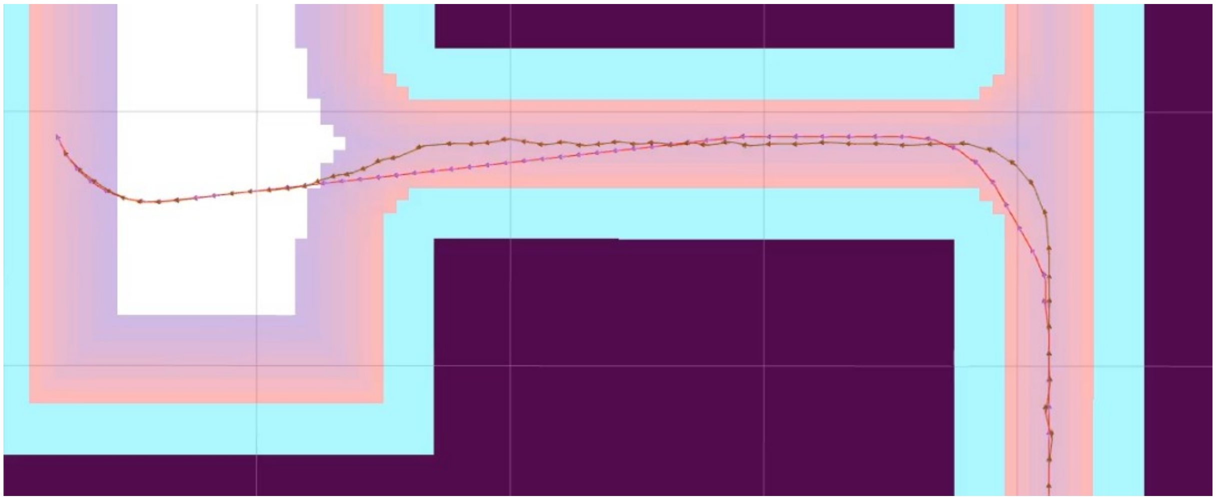
The environment setup for this benchmark is the same as was expressed in the "[Comparison of effects probability]" chapter. During this benchmark `w_smooth` residual coefficient to be varied while the rest were set to production default: `w_cost=0.015, w_curve=30.0 w_smooth=0.0`.

The variation of the `w_smooth` was chosen to confirm the minimal choice of new value, and was passed in range from its production value 15K up to proposed in previous chapters 2M. Each oscillation effect was ranged by its strength by following gradations:

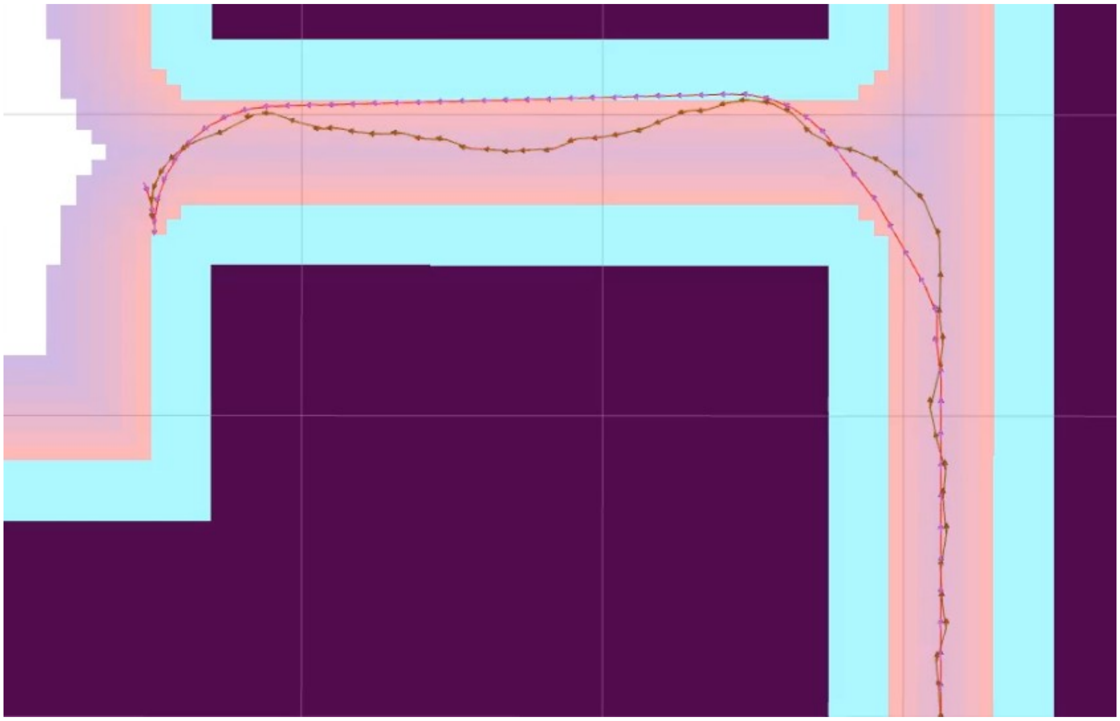
0. No notable oscillations were found
1. Low oscillation effect: some individual almost imperceptible deviations from the desired smooth trajectory are presented, like depicted below (red – raw path, brown – smoothed one):



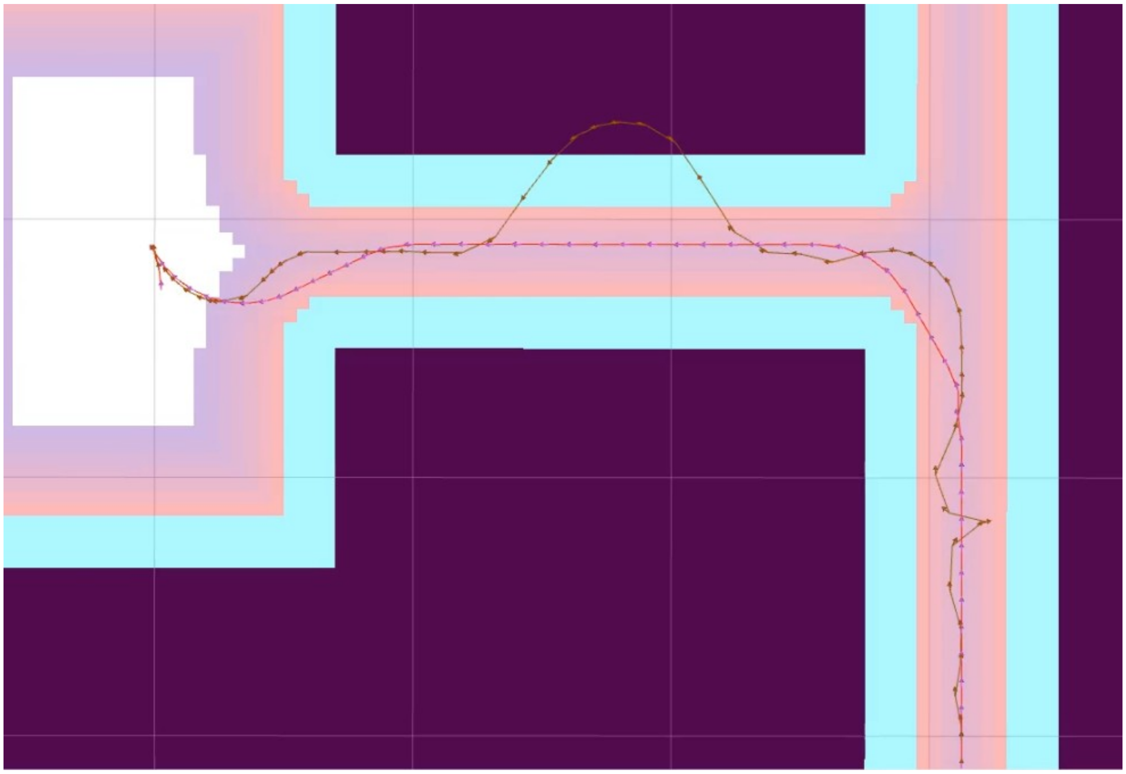
2. Moderate oscillation effect: scatters are seem and notably spoil the quality of smoothed path:



3. High oscillation effect: scatters with high amplitude appear on whole smoothed path are impacting the quality of whole smoothed path:



4. Extremely high effect: most complex cases, like presented in the [Background]() and [Path planner effect check]() chapters, making the smoother work to be completely unacceptable. Another example of extremely high effect could be also find at the picture below:



Produced paths observations through 100 randomly generated paths for different `w_smooth` residual coefficient could be found in the table below.

w_smooth	**15K**	**300K**	**1M**	**2M**	
Oscillation score	77	44	52	33	
Oscillation effect	Number of experiments				
None	64	67	57	68	← Good
Low	7	27	34	31	← Noise
Moderate	20	1	9	1	← Undesirable
High	6	5	0	0	← Impacting
Extremely high	3	0	0	0	← Catastrophic

Here the oscillation score – is the summary of oscillation effects marks calculated by following mapping: `None→0, Low→1, Moderate→2, High→3, Extremely high→4`.
For example, the oscillation score for `w_smooth = 15K: 7*1 + 20*2 + 6*3 + 3*4 =` **`77`**.

Through this experimentation, `w_smooth = 15K` shown lots of moderate, high and even 3 extremely high oscillations in the Narrow World, so making it to be inapplicable for the best candidate selection.

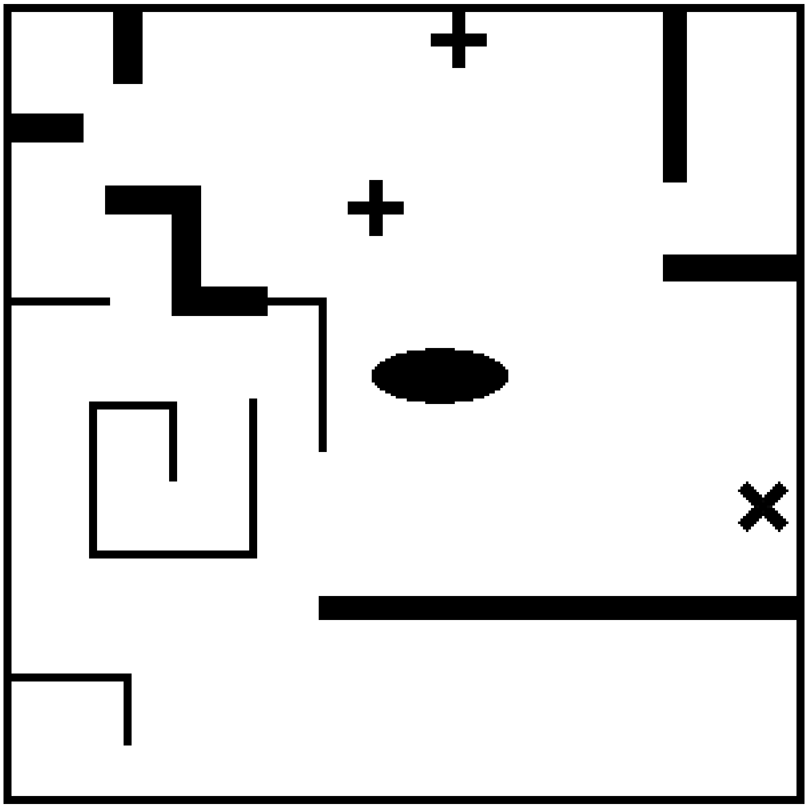
`w_smooth = 300K` shown much more stable results. However, the numbers of high impacting oscillations (most of these cases are migrated from 15K's "extreme" ones) making this selection to be inappropriate.

`w_smooth = 1M` is showing pretty satisfying results. But due to large number of moderate oscillations will be discarded in favor of 2M.

`w_smooth = 2M` is showing the best overall picture with no high or extremely high oscillations, and one moderate noted during experimentation. 2M is also having the best oscillation score, which making it to be the choice for the optimal set for Constrained Smoother.

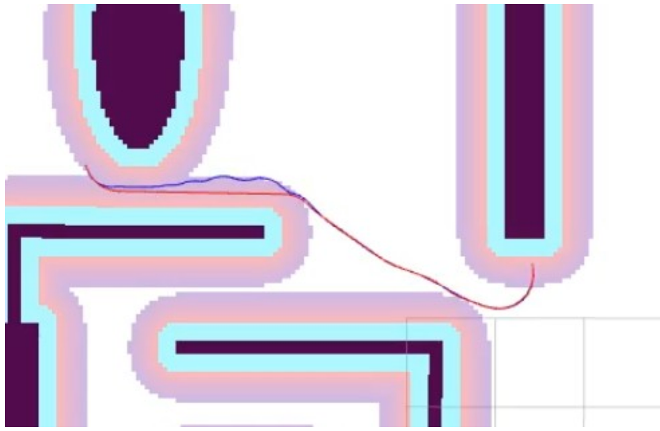
Smoothers World

To confirm the integrity of `w_smooth` coefficient choice, the experimentation should proceed in different environments. Previous Narrow World was the example with narrow hallways and spaces, almost completely covered by inflation. Despite it, Smoothers World – is a large-scaled world with lots of open spaces with relatively low percent of inflated spaces, that allowing path planners and Constrained Smoother to operate more straightforward. The Smoothers World could be found in a `tools/smoother_benchmarking/maps` in a Nav2 stack as a standard world for path smoothers benchmark:

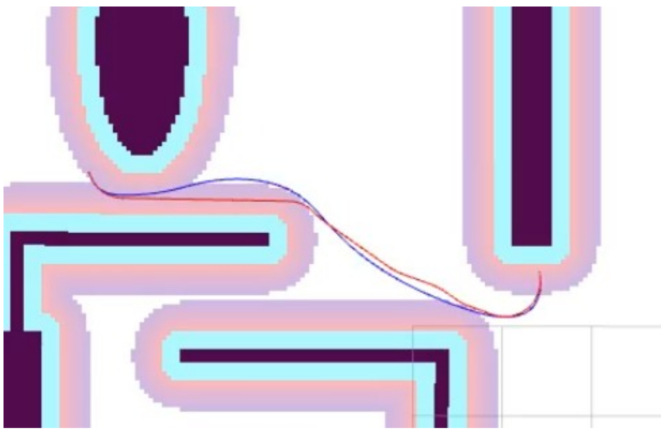


Smoothers World (300x300 cells = 15m x 15m)

The experimentation was proceeded by the same scheme as in previous sub-chapter, and showing no oscillations for the 2M case comparing to some noted cases when `w_smooth=15K`:



Oscillation in the Smoothers World of Constrained Smoother operating with w_smooth=15K. Raw path depicted in red, smoothed path – in blue.



Oscillation in the Smoothers World of Constrained Smoother operating with w_smooth=2M. Raw path depicted in red, smoothed path – in blue.

Random paths benchmarking

During the experimentation, the effect of `w_smooth` coefficient change was measured on path smoothers benchmarking suite, placed in `tools/smoother_benchmarking` in Nav2 stack.

All parameters during the benchmark were set to default, for both for SmacHybridA* path planner and Constrained Smoother. The experimentation was proceeded in used above Smoothers World on 1000 randomly generated start-goal pose pairs. `w_smooth` coefficient value as always, was varied. The results of benchmarking could be found in the tables below:

For `REEDS_SHEPP` motion model:

Method	**Time (ms)**	**Path length (m)**	**Average cost**	**Max cost**	**Smoothness (x100)**	**Average turning rad (m)**
SmacHybrid	9.36	10.60	18.11	142.02	81.01	0.66
CS w_smooth=15K	18.57	10.67	8.20	101.77	128.71	2.17
CS w_smooth=300K	20.29	10.60	11.31	115.40	93.30	2.25
CS w_smooth=1M	20.31	10.58	15.85	129.74	88.47	2.26
CS w_smooth=2M	20.56	10.58	16.86	135.12	87.42	2.28

And for `DUBIN` motion model:

Method	**Time (ms)**	**Path length (m)**	**Average cost**	**Max cost**	**Smoothness (x100)**	**Average turning rad (m)**
SmacHybrid	21.67	11.11	18.27	142.88	67.46	0.66
CS w_smooth=15K	20.22	11.17	8.02	98.62	110.23	2.22
CS w_smooth=2M	21.81	11.11	17.29	138.43	75.68	2.27

Increasing the `w_smooth` value leads to the average path length decrease, tuning radius increase and path smoothness decrease which is definitely an improvement of the smoother metrics. However the other side of the coin is an effect of inevitable increase of the average and maximum cost from higher `w_smooth` values, which are inherently from the equations could not be nullified. Despite on this effect, maximum and average costs are still lower than initially produced by SmacHybridA* planner, which allows the `w_smooth=2M` to be finally chosen for the production.

Conclusion

To consider the root cause of path oscillation behavior it is important to cross off the effects that are could not be related to path oscillation, remaining only the one which will be the root cause of problem. Analysis shown that the path oscillation is not related to any spurious/extreme iteration or top-limiting of max Ceres algorithm iterations number. Also, the oscillation effect does not depend of Ceres solver function minimum tolerance or gradient of descent.

Thus, path oscillations are caused by Ceres problem function `F` residuals contribution. The residuals could be divided into the negative- and positive-effect producing parts of the equation. Negative effect-producing parts are **`w_cost`** and **`w_curve`** while positive effect producing are: **`w_smooth`** and **`w_dist`**.

Source code analysis of parts related to **`w_cost`** and **`w_curve`** residual calculation functions was shown no abnormalities. Since the **`w_cost`** and **`w_curve`** parts of the Ceres problem function `F` have no knowledge about path continuity and integrity, they should be used along with **`w_smooth`** and **`w_dist`** to produce optimal smoothed paths. Only if both groups of residuals are working together, we could obtain the necessary performance of the Constrained Smoother algorithm.

During the numerous tests and benchmarks under different environments (on small and large maps, narrow- and wide-spaced) it was found and proved the optimal set of the residual coefficients causing Constrained Smoother to produce the rational path without oscillation effects:

Residual coefficient	Standard value change
`w_smooth`	15,000 → 2,000,000
`w_cost`	0.015
`w_curve`	30.0
`w_dist`	0.0

Having `w_smooth` 9 order of magnitude greater than `w_cost` and 6 order of magnitude greater than `w_curve` relation showing that Constrained Smoother still will produce the path in respect to low-costs areas and curves.