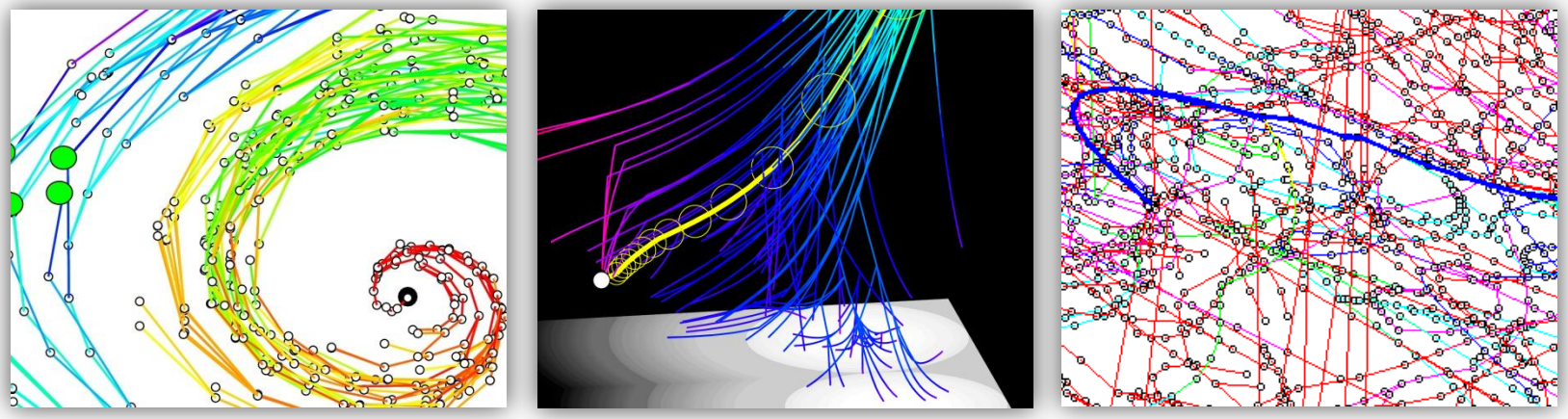
LQR-RRT*: Optimal Sampling-Based Motion Planning with Automatically Derived Extension Heuristics



Alejandro Perez

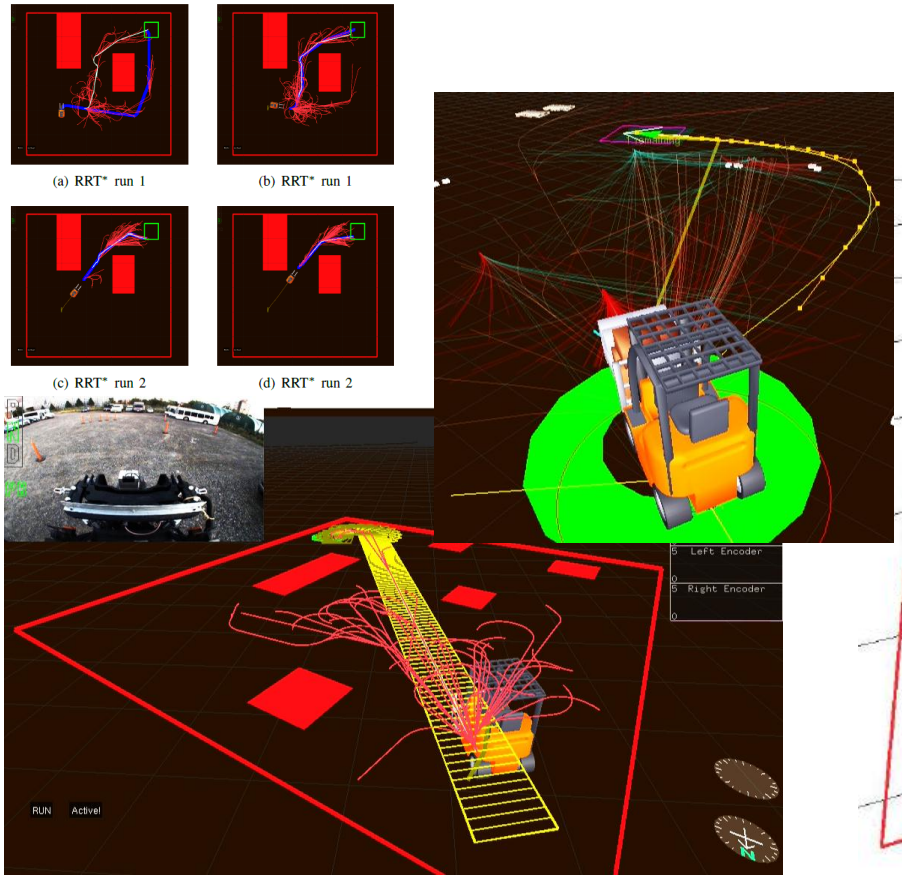
Robert Platt Jr., George Konidaris,

Leslie Kaelbling and Tomas Lozano-Perez

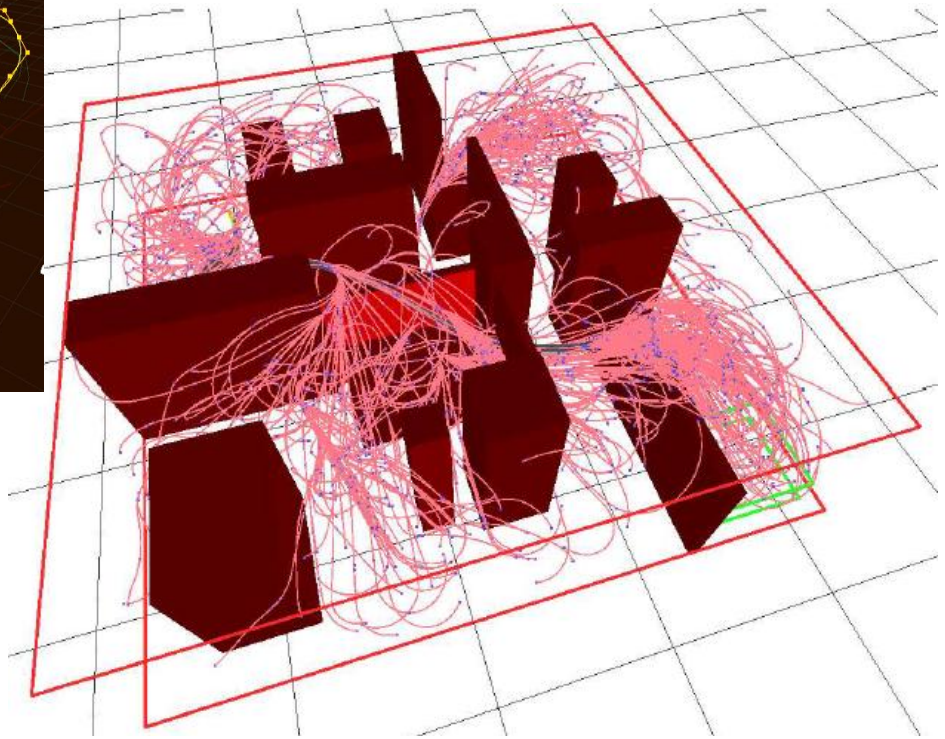
MIT/CSAIL

IEEE International Conference on Robotics and Automation 2012

Motion Planning with Dynamics



Karaman, Walter, Perez, Frazzoli, and Teller, "Anytime Motion Planning using the RRT*," in IEEE International Conference on Robotics and Automation (ICRA), 2011.



Karaman and Frazzoli, "Optimal Kinodynamic Motion Planning using Incremental Sampling-based Methods," in IEEE Conference on Decision and Control (CDC), Atlanta, GA, 2010.

Motion Planning with Dynamics

Torque-limited pendulum

Two-dimensional state space, $(\theta, \dot{\theta})$

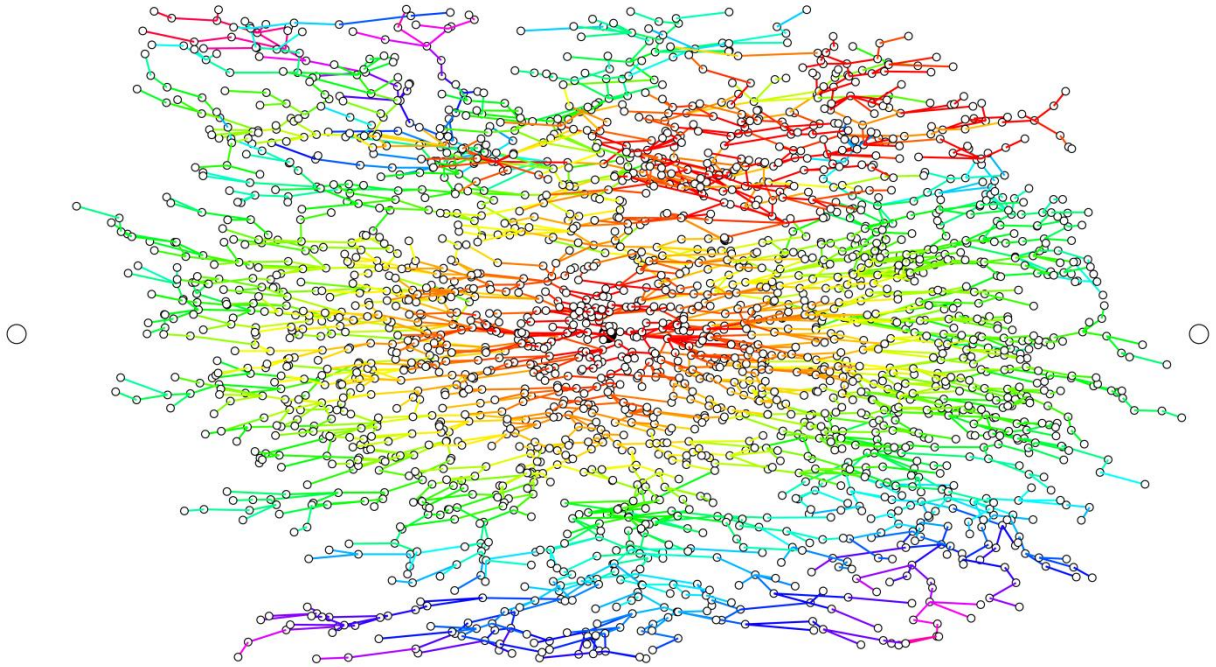
$$\ddot{\theta} = u - b\dot{\theta} - g\cos(\theta),\ g = 9.81,\ b = 0.1,\ u \in [-3, 3]$$

Motion Planning with Dynamics

Torque-limited pendulum

Two-dimensional state space, $(\theta, \dot{\theta})$

$$\ddot{\theta} = u - b\dot{\theta} - g\cos(\theta),\ g = 9.81,\ b = 0.1,\ u \in [-3, 3]$$



Motion Planning with Dynamics

Torque-limited pendulum

Two-dimensional state space, $(\theta, \dot{\theta})$

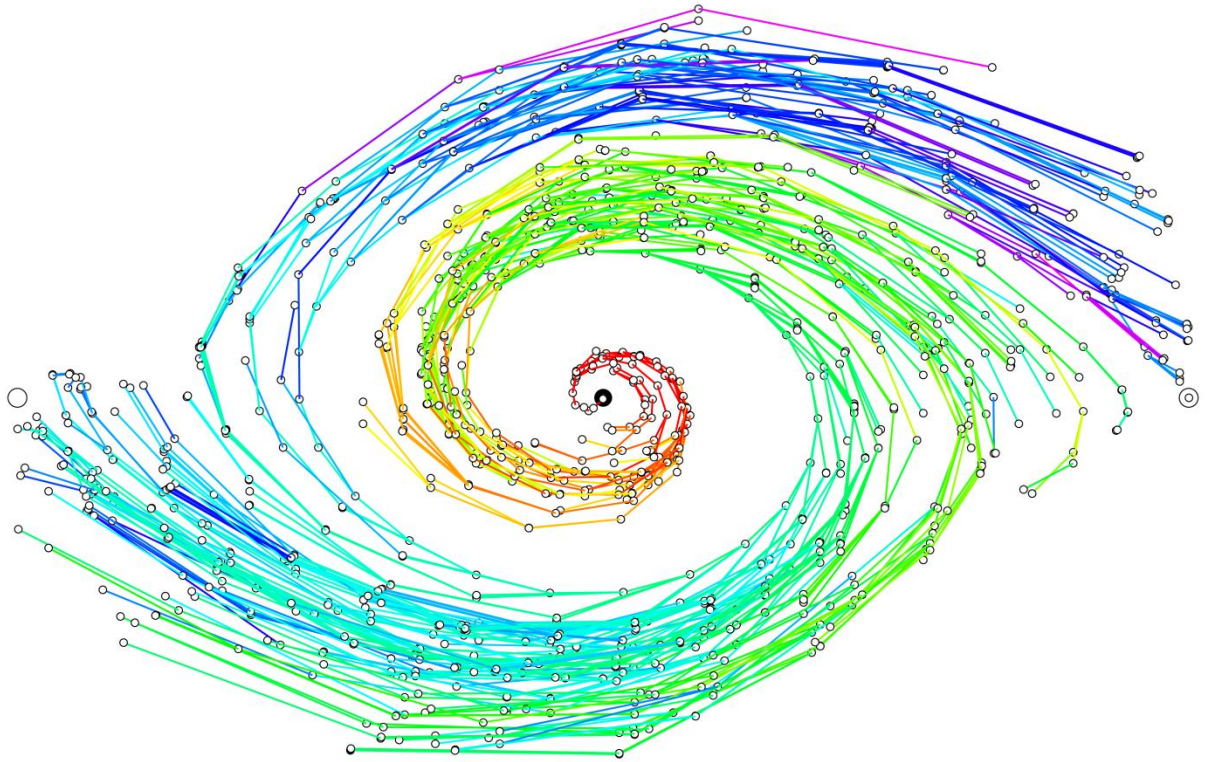
$$\ddot{\theta} = u - b\dot{\theta} - g\cos(\theta),\ g = 9.81,\ b = 0.1,\ u \in [-3, 3]$$

Motion Planning with Dynamics

Torque-limited pendulum

Two-dimensional state space, $(\theta, \dot{\theta})$

$$\ddot{\theta} = u - b\dot{\theta} - g\cos(\theta),\ g = 9.81,\ b = 0.1,\ u \in [-3, 3]$$



Motion Planning with Dynamics

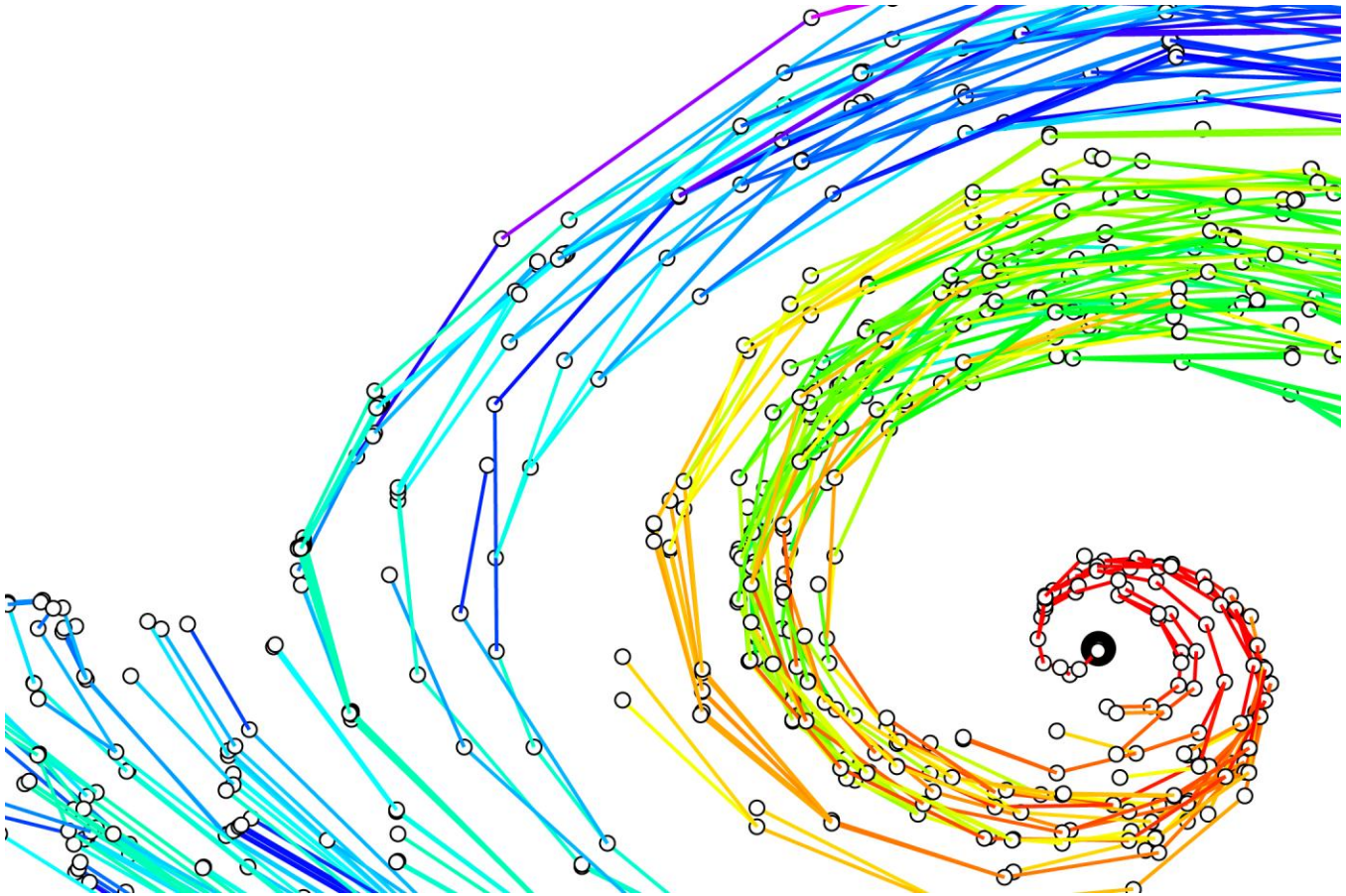
Torque-limited pendulum

Two-dimensional state space, $(\theta, \dot{\theta})$

$$\ddot{\theta} = u - b\dot{\theta} - g\cos(\theta),\ g = 9.81,\ b = 0.1,\ u \in [-3, 3]$$

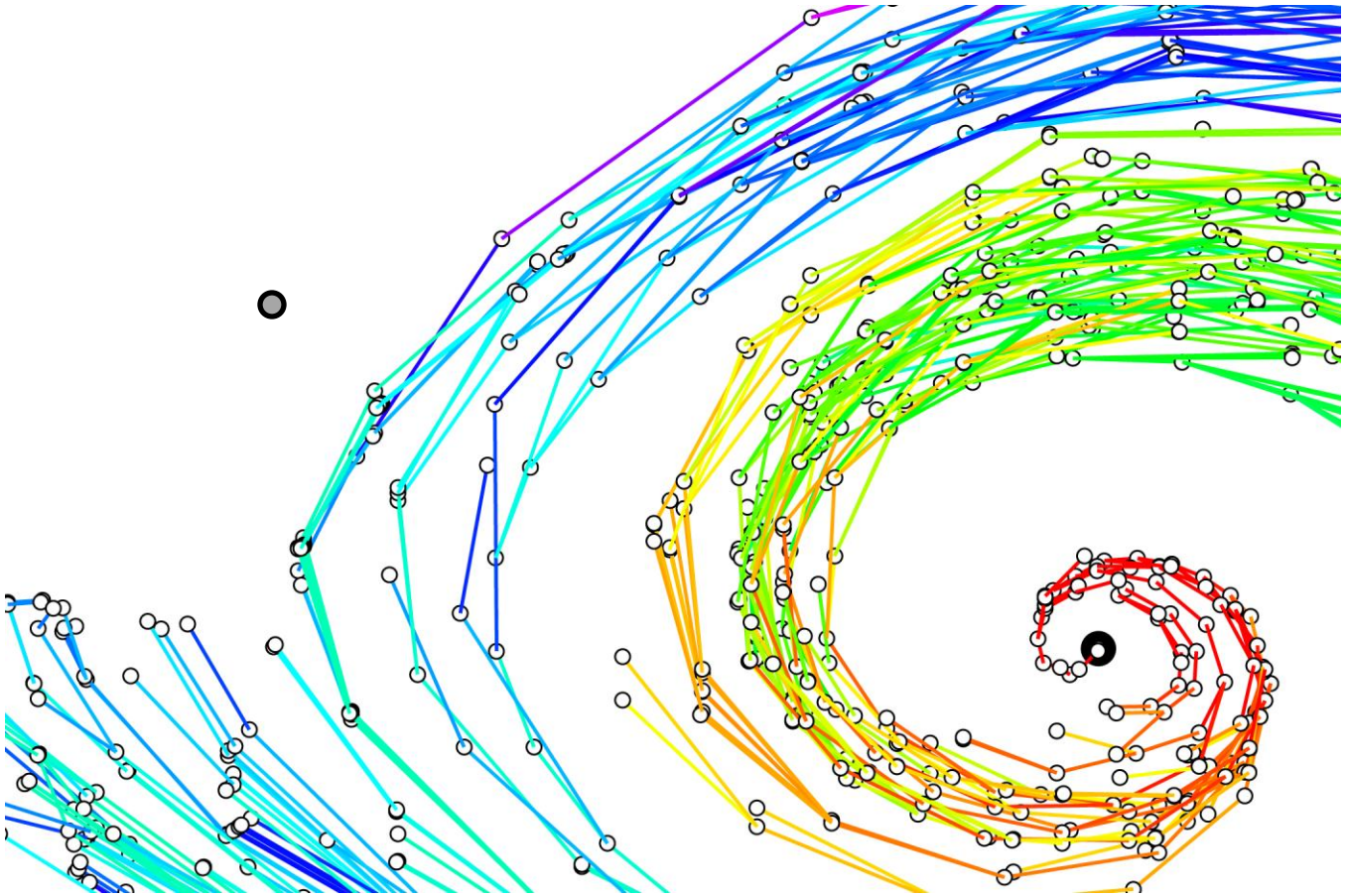
Motion Planning with Dynamics

Cost Metric



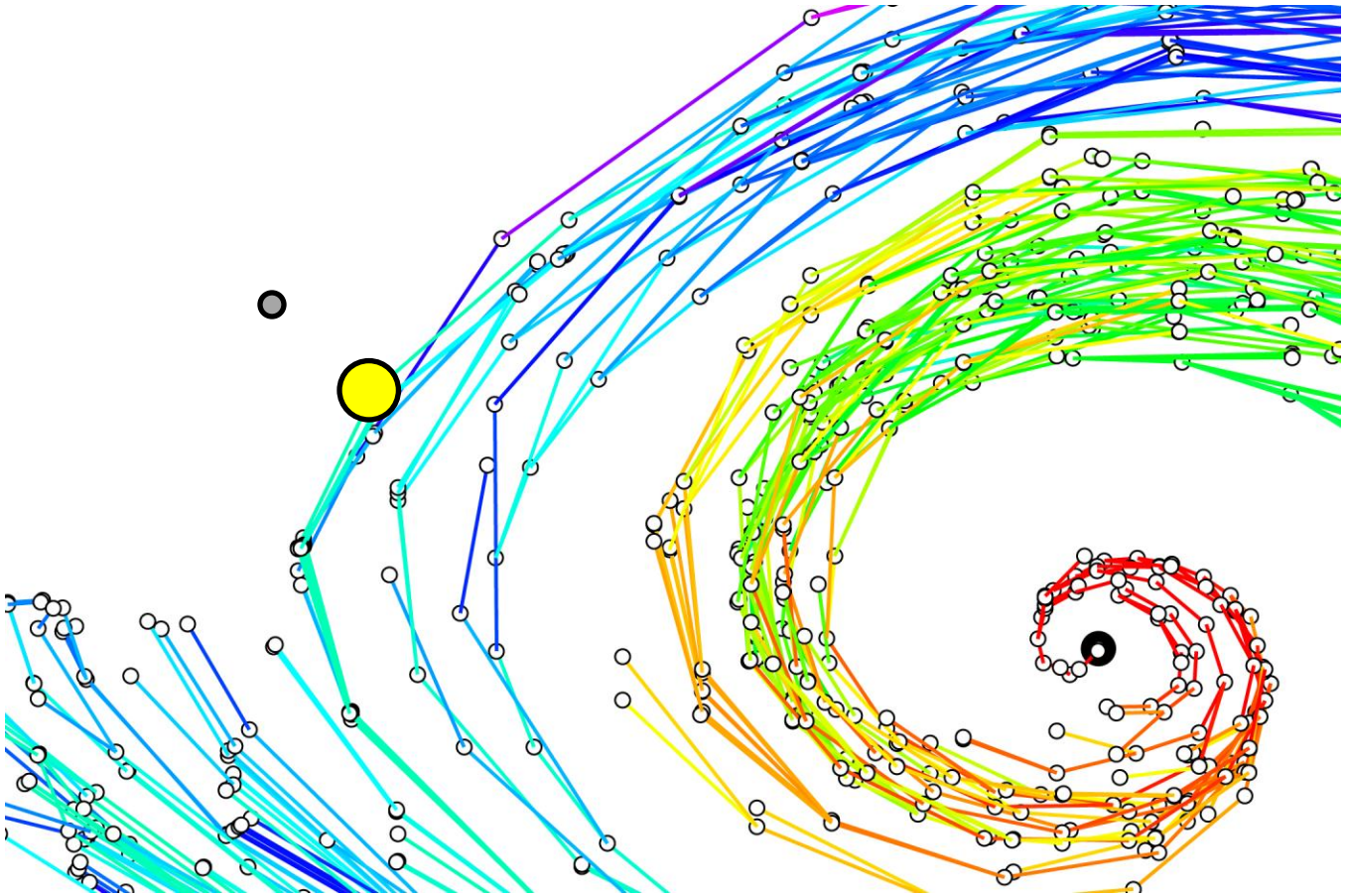
Motion Planning with Dynamics

Cost Metric



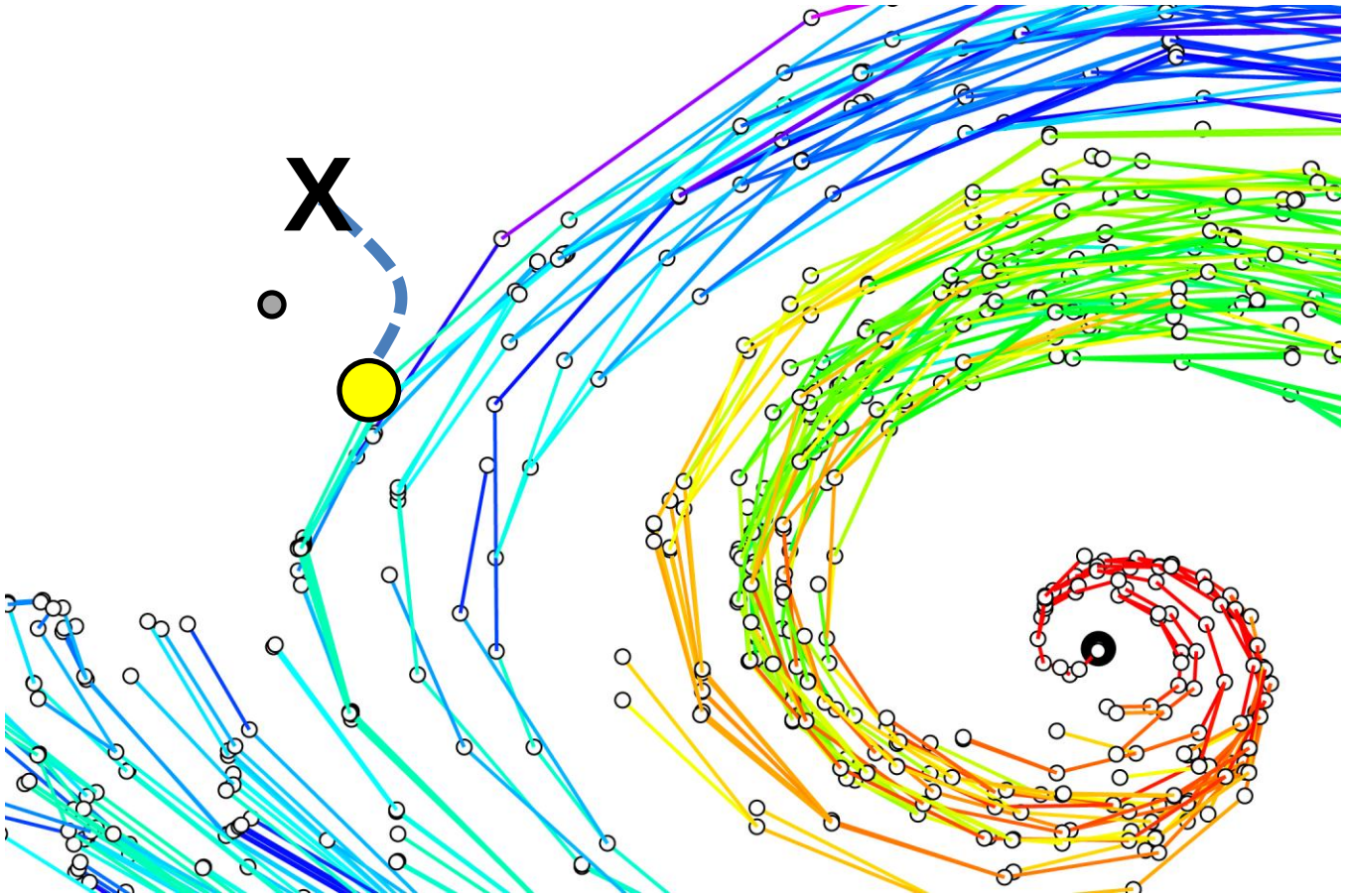
Motion Planning with Dynamics

Cost Metric



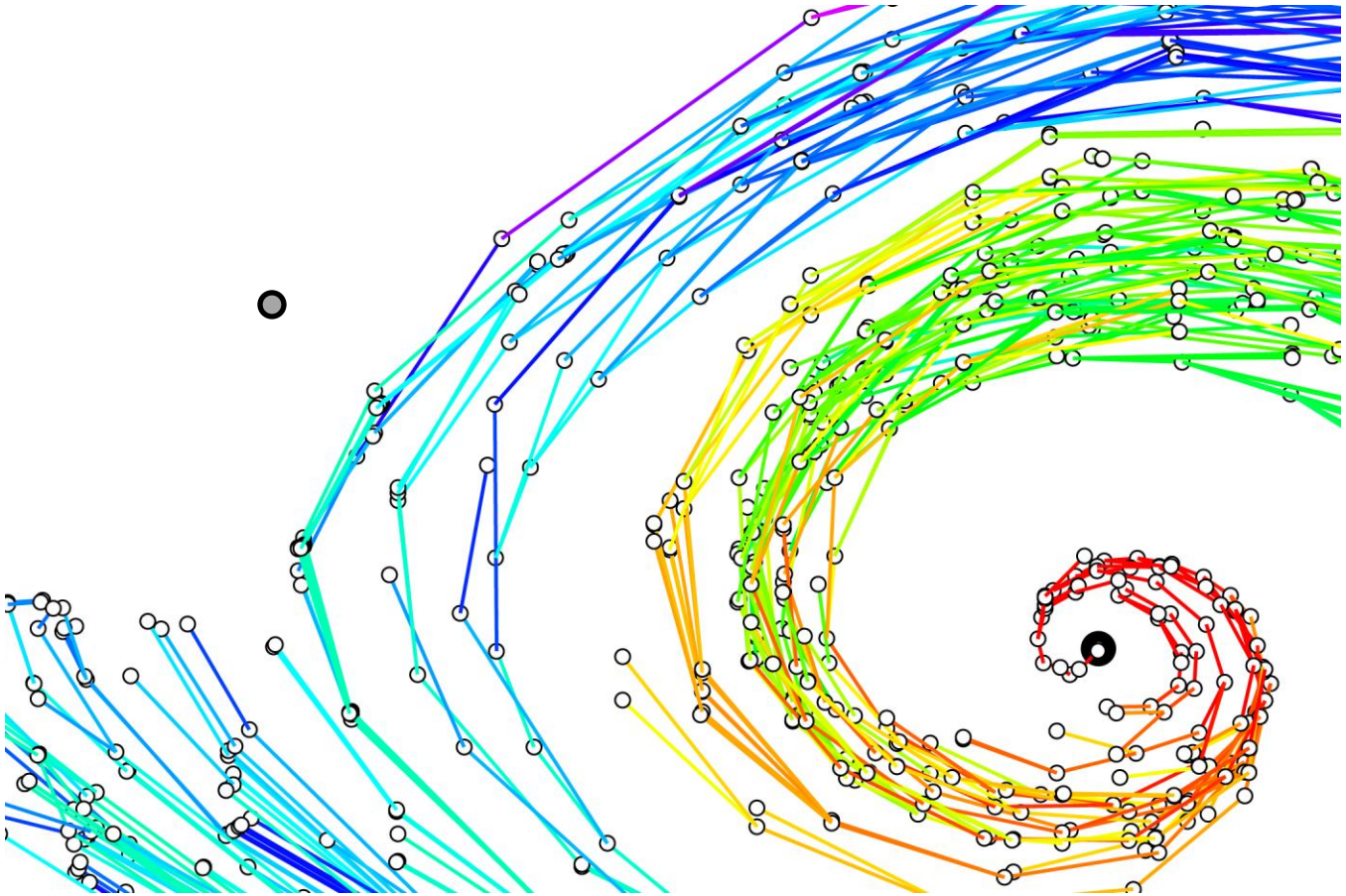
Motion Planning with Dynamics

Cost Metric



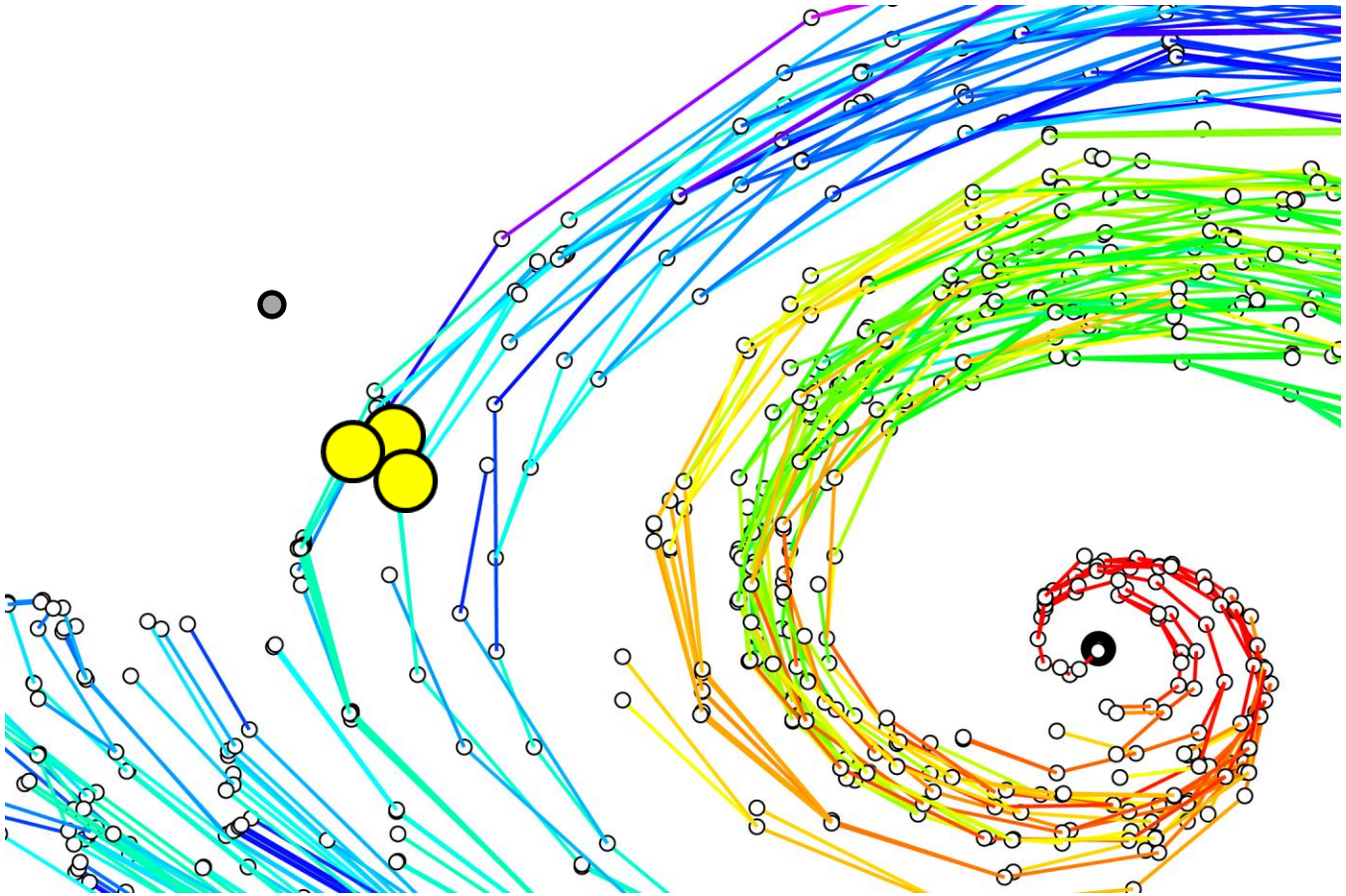
Motion Planning with Dynamics

Cost Metric



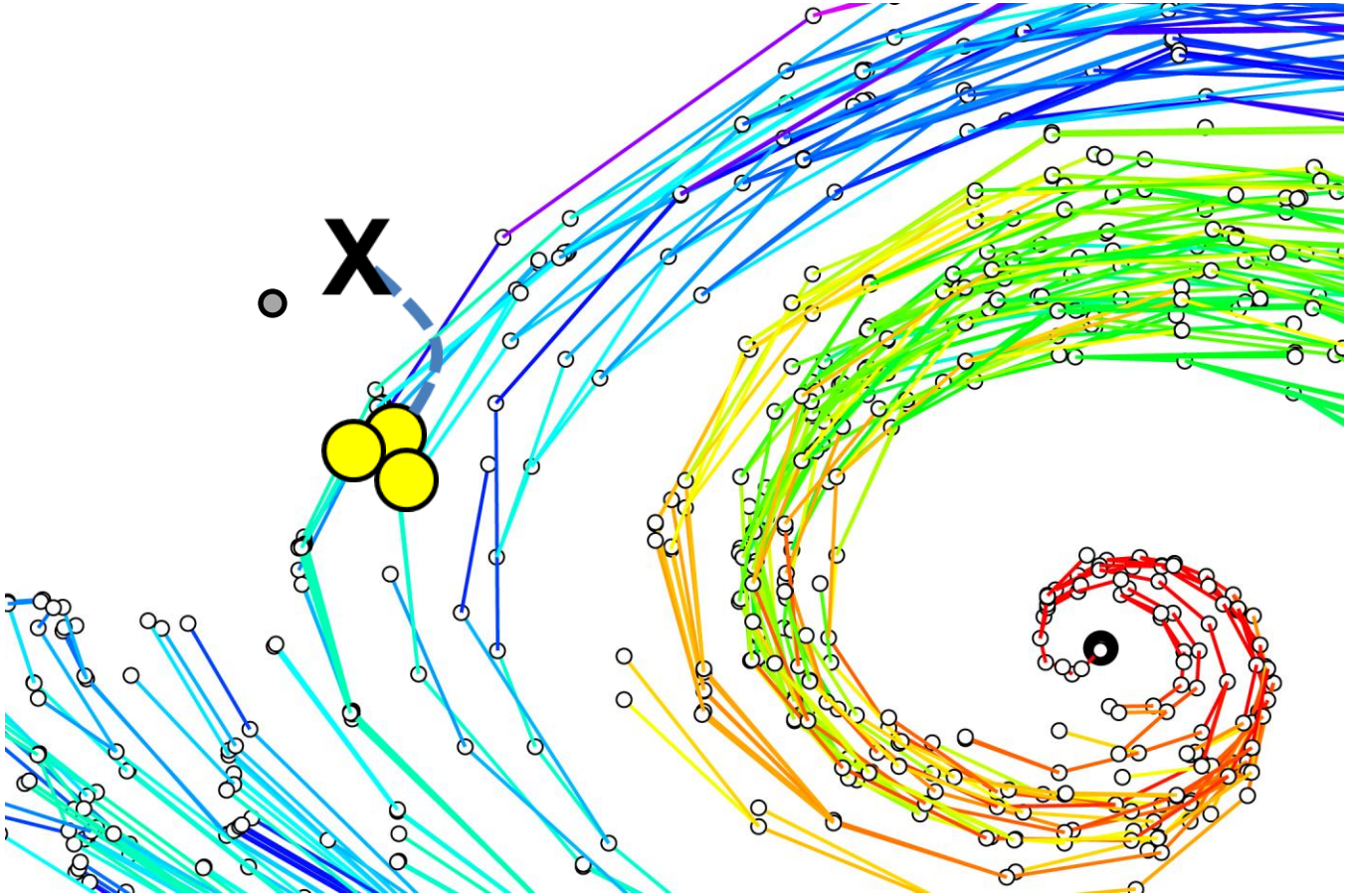
Motion Planning with Dynamics

Cost Metric



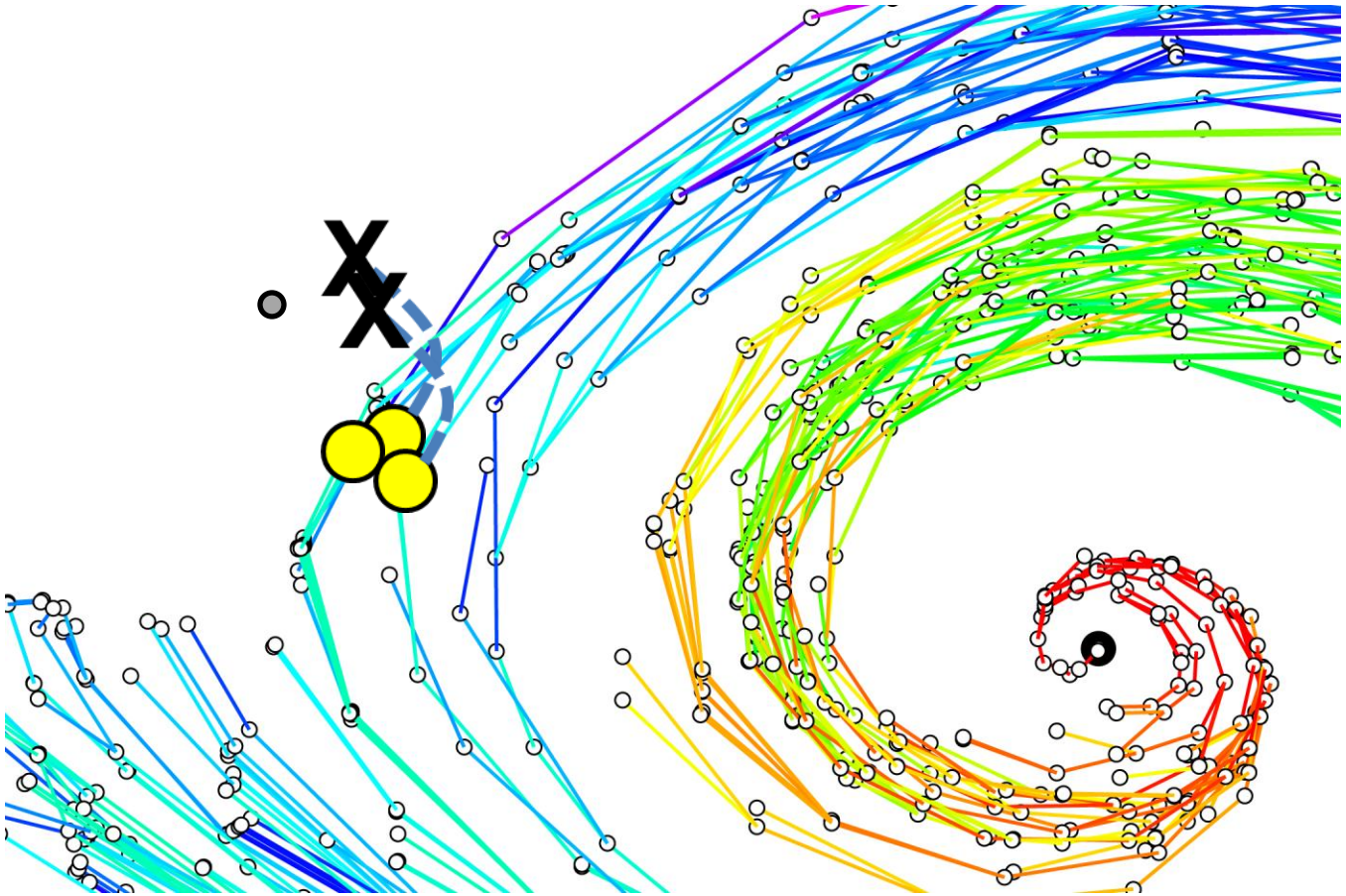
Motion Planning with Dynamics

Cost Metric



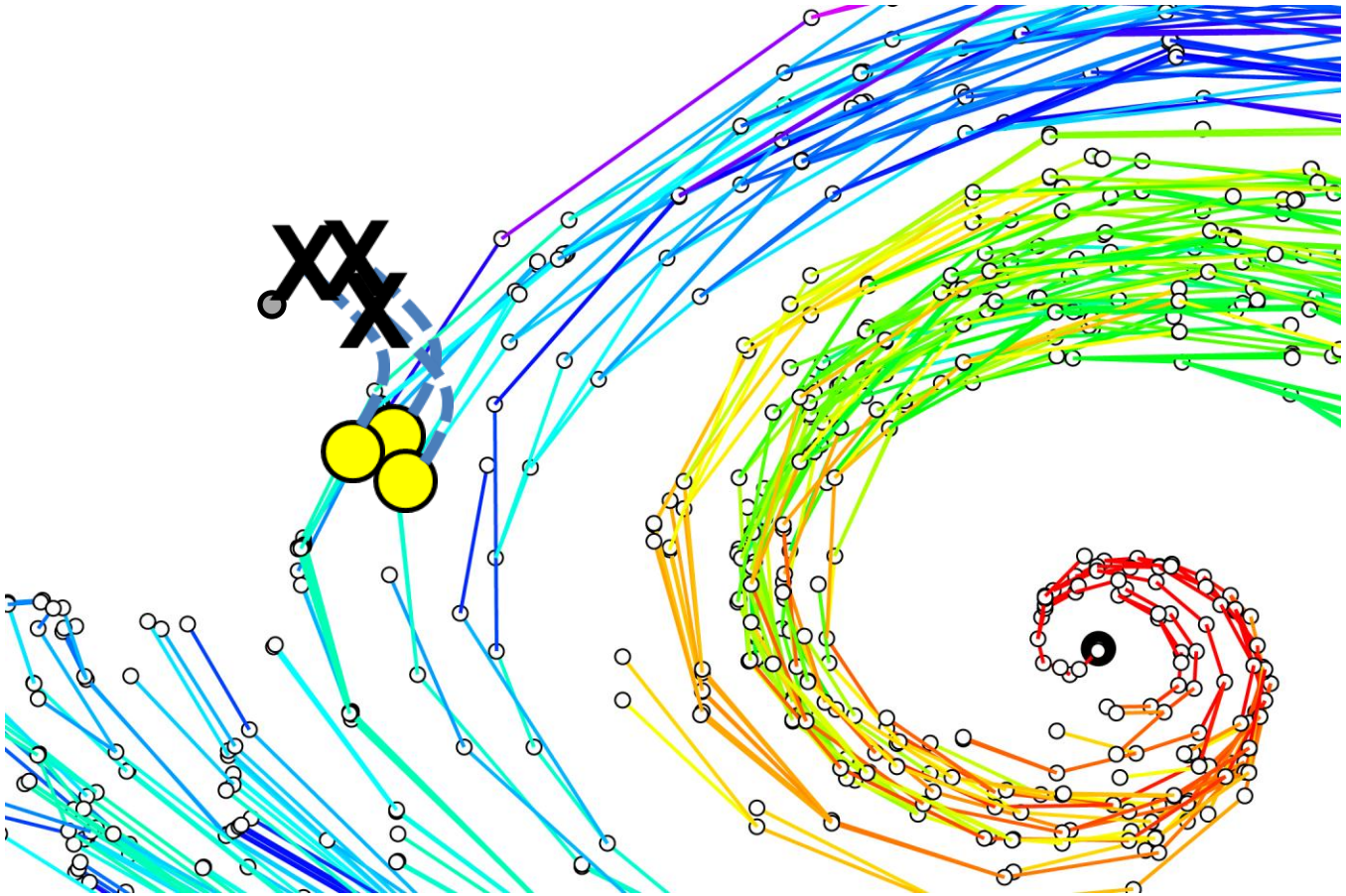
Motion Planning with Dynamics

Cost Metric



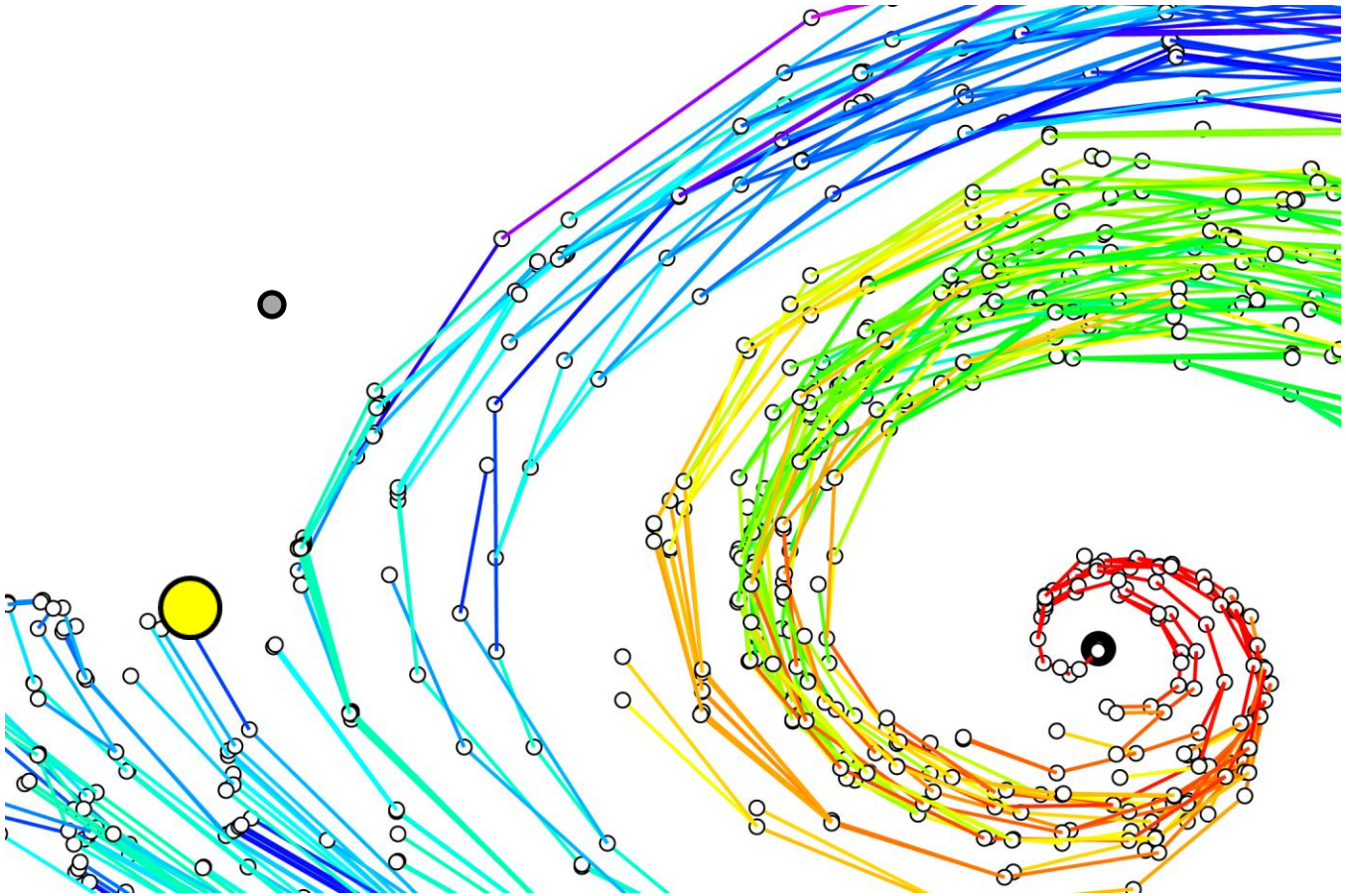
Motion Planning with Dynamics

Cost Metric



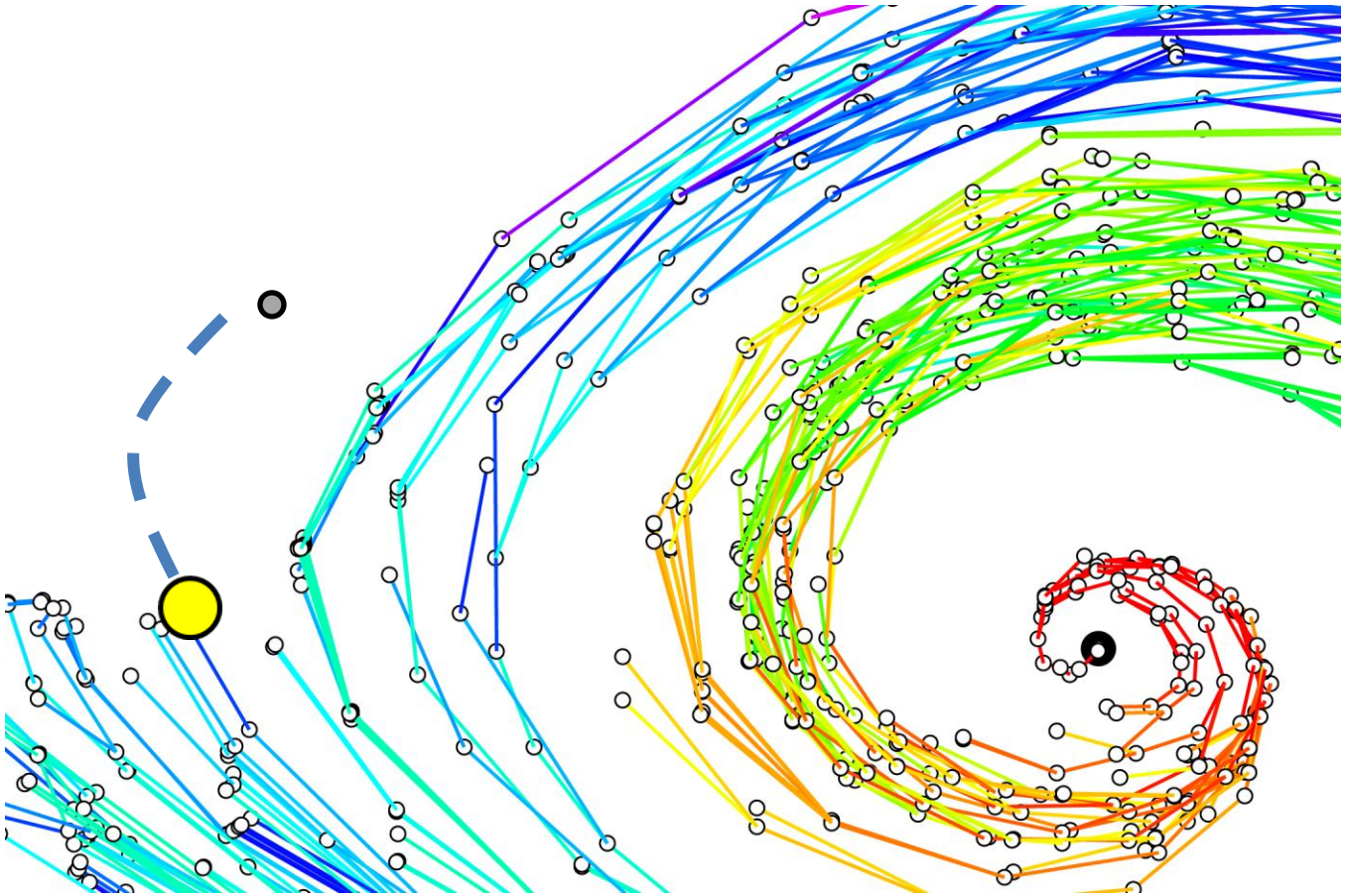
Motion Planning with Dynamics

Steer Function



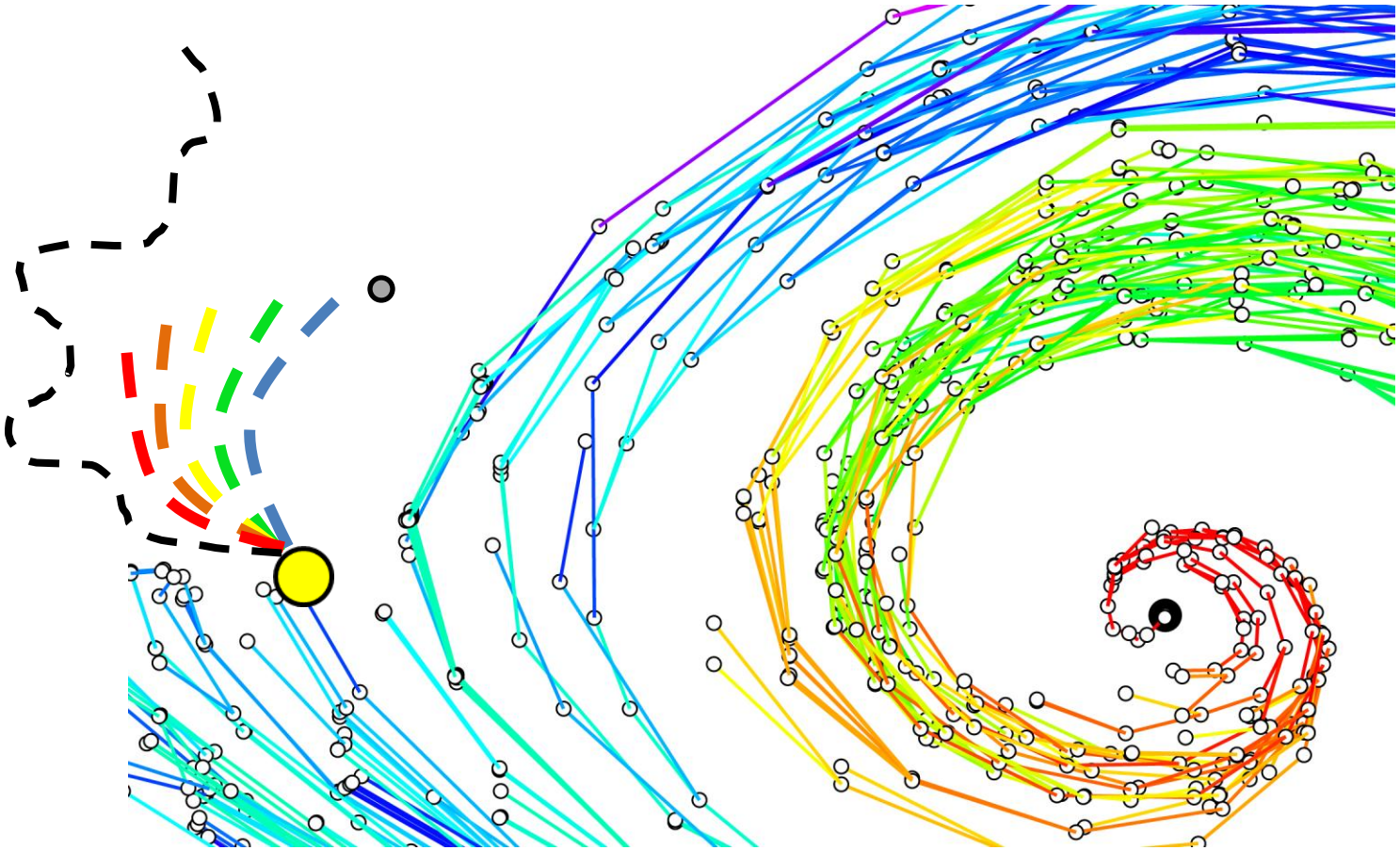
Motion Planning with Dynamics

Steer Function



Motion Planning with Dynamics

Steer Function



Motion Planning with Dynamics

Relevant Literature

Performance of RRT-based algorithms is sensitive to the distance metric used
S. M. Lavalle, "From dynamic programming to RRTs: Algorithmic design of feasible trajectories," in Control Problems in Robotics. Springer-Verlag, 2002.

RRTs efficiently explore the state space only when this metric reflects the true cost-to-go
P. Cheng and S. M. Lavalle, "Reducing metric sensitivity in randomized trajectory design," in In IEEE International Conference on Intelligent Robots and Systems, 2001, pp. 43–48.

Cost-to-go functions can be based on optimal control for linearized systems
S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," International Journal of Robotics Research, vol. 20, no. 5, pp. 378–400, May 2001.

Asymptotic optimality of RRT* is shown
S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," International Journal of Robotics Research, June 2011.

Cost-to-go pseudometric based on LQR is derived
E. Glassman and R. Tedrake, "A quadratic regulator-based heuristic for rapidly exploring state space," in Proceedings of the IEEE International Conference on Robotics and Automation, May 2010.

Sufficient conditions for optimality of RRT* for kinodynamics are presented
Karaman and Frazzoli, "Optimal Kinodynamic Motion Planning using Incremental Sampling-based Methods," in IEEE Conference on Decision and Control (CDC), Atlanta, GA, 2010. .

Approach Overview

- Asymptotically optimal incremental sampling-based approach
- Linearize process dynamics about sampled state
- Quadratic cost-to-go metric
- Steer with approximated locally optimal policy

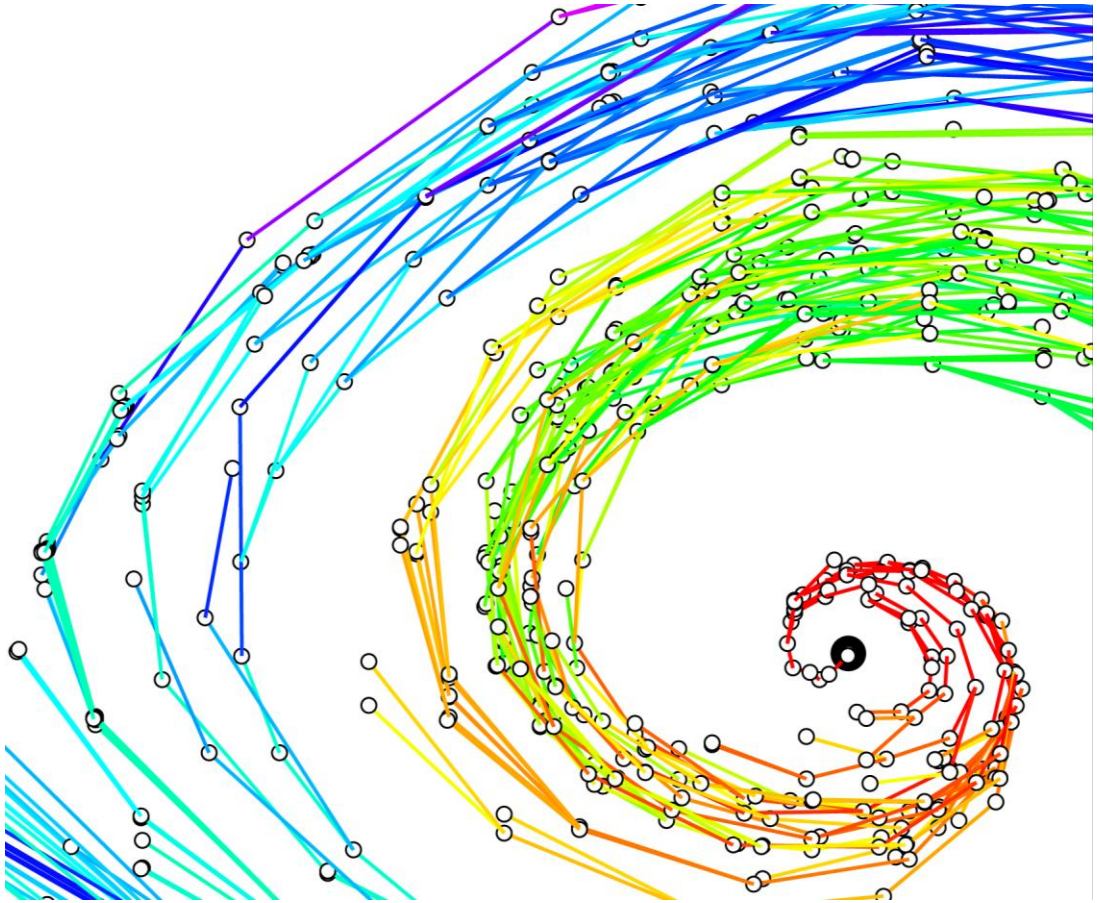
Linear Quadratic Regulation (LQR)

We will use the notation

$$[\mathbf{K}, \mathbf{S}] = \mathbf{LQR}(\mathbf{A}, \mathbf{B}, \mathbf{Q}, \mathbf{R}),$$

to denote the function that calculates the LQR gain matrix, K, and the cost matrix, S.

LQR-RRT*



Algorithm 4: $\texttt{LQR} - \texttt{RRT}^*((V, E), N)$

```
for i = 1, ..., N do
    x_rand ← Sample;
    x_nearest ← LQRNearest(V, x_rand);
    x_new ← LQRSteer(x_nearest, x_rand);
    X_near ← LQRNear(V, x_new);
    (x_min, σ_min) ← ChooseParent(X_near, x_new);
    if CollisionFree(σ) then
        X ← X ∪ {x_new};
        E ← E ∪ {(x_min, x_new)};
        (V, E) ← Rewire( (V, E), X_near, x_new );
return G = (V, E);
```

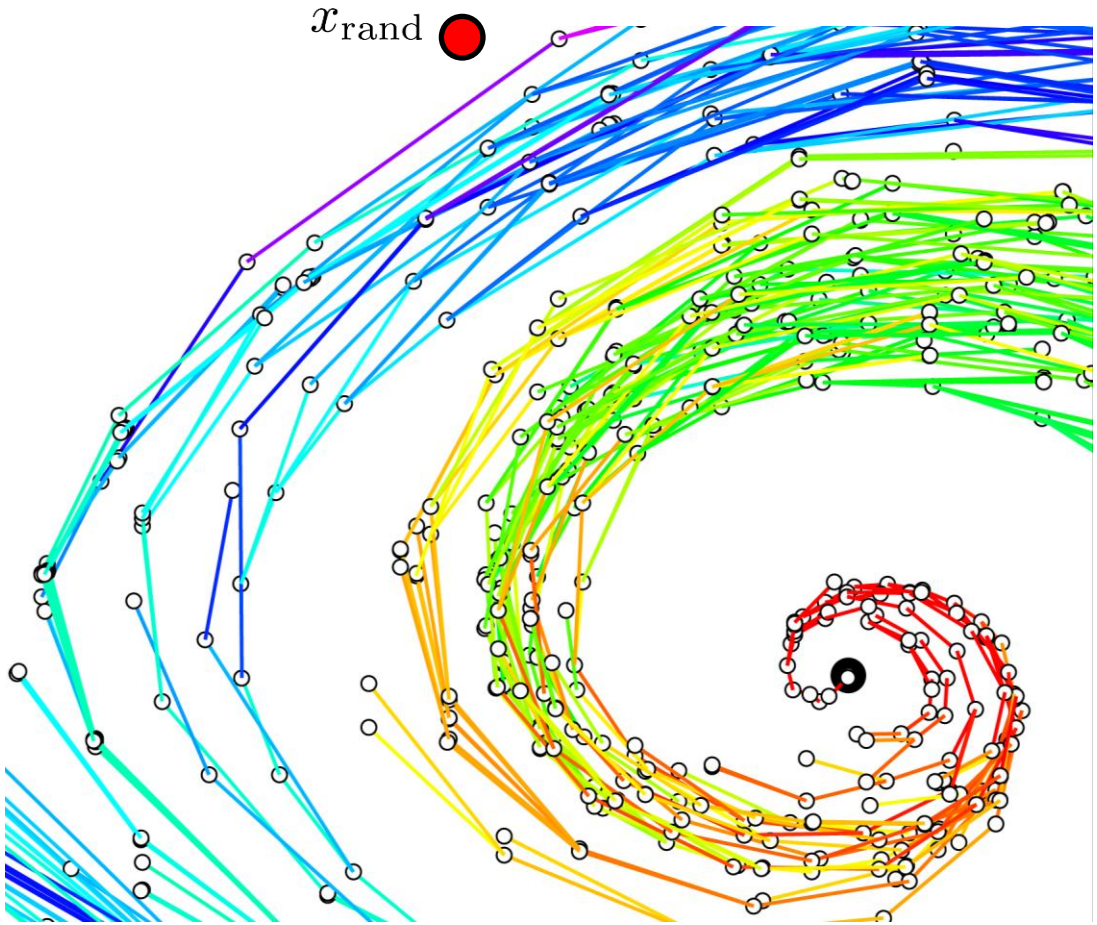
Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

```
minCost ← ∞;  x_min ← NULL;  σ_min ← NULL;
for x_near ∈ X_near do
    σ ← LQRSteer(x_near, x_new);
    if Cost(x_near) + Cost(σ) < minCost then
        minCost ← Cost(x_near) + Cost(σ);
        x_min ← x_near;  σ_min ← σ;
return (x_min, σ_min);
```

Algorithm 6: $\texttt{Rewire}(\,(V, E), X_{\text{near}}, x_{\text{new}}\,)$

```
for x_near ∈ X_near do
    σ ← LQRSteer(x_new, x_near) ;
    if Cost(x_new) + Cost(σ) < Cost(x_near) then
        if CollisionFree(σ) then
            x_parent ← Parent(x_near);
            E ← E \ {x_parent, x_near};
            E ← E ∪ {x_new, x_near};
return (V, E);
```

LQR-RRT*



Algorithm 4: $\texttt{LQR} - \texttt{RRT}^*((V, E), N)$

```
for i = 1, ..., N do
    x_rand ← Sample;
    x_nearest ← LQRNearest(V, x_rand);
    x_new ← LQRSteer(x_nearest, x_rand);
    X_near ← LQRNear(V, x_new);
    (x_min, σ_min) ← ChooseParent(X_near, x_new);
    if CollisionFree(σ) then
        X ← X ∪ {x_new};
        E ← E ∪ {(x_min, x_new)};
        (V, E) ← Rewire( (V, E), X_near, x_new );
return G = (V, E);
```

Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

```
minCost ← ∞;  x_min ← NULL;  σ_min ← NULL;
for x_near ∈ X_near do
    σ ← LQRSteer(x_near, x_new);
    if Cost(x_near) + Cost(σ) < minCost then
        minCost ← Cost(x_near) + Cost(σ);
        x_min ← x_near;  σ_min ← σ;
return (x_min, σ_min);
```

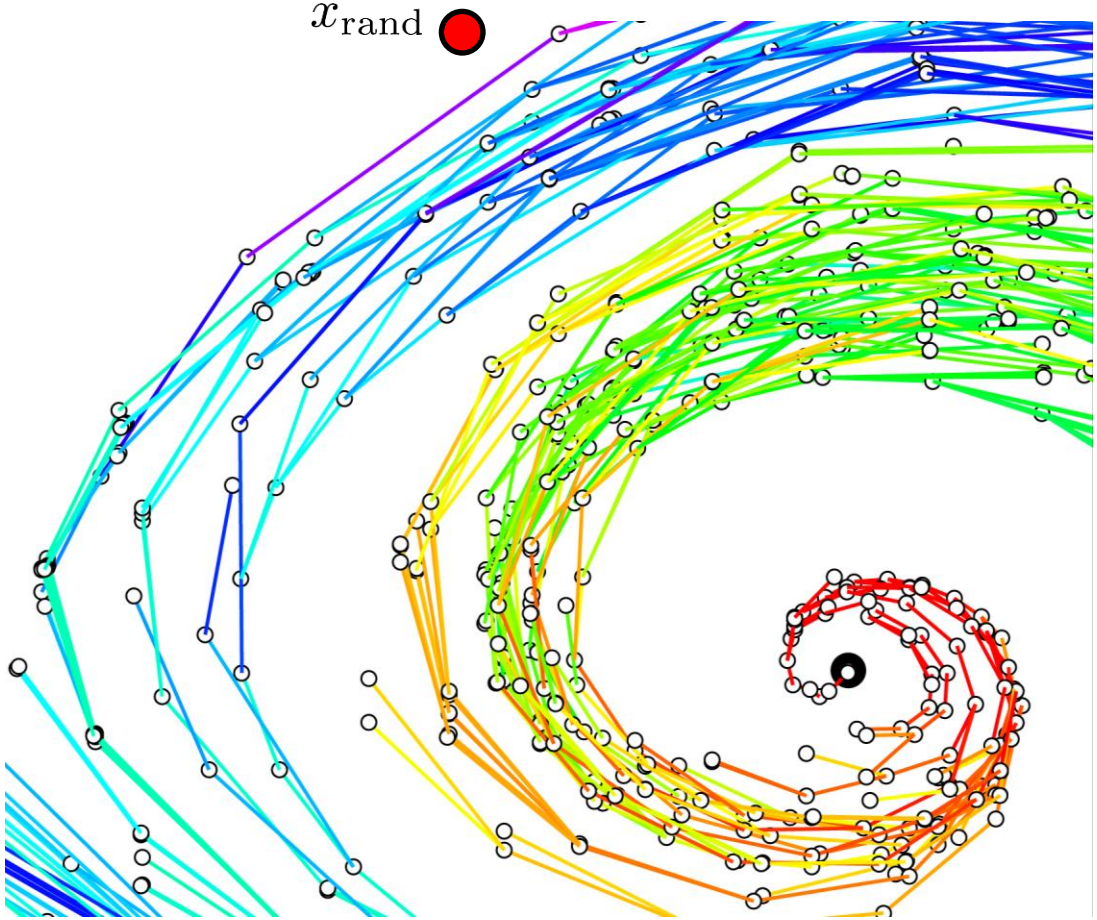
Algorithm 6: $\texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$

```
for x_near ∈ X_near do
    σ ← LQRSteer(x_new, x_near) ;
    if Cost(x_new) + Cost(σ) < Cost(x_near) then
        if CollisionFree(σ) then
            x_parent ← Parent(x_near);
            E ← E \ {x_parent, x_near};
            E ← E ∪ {x_new, x_near};
return (V, E);
```

LQR-RRT*

$$[K(x), S(x)] = LQR(A(x,0), B(x,0), Q, R).$$

$$x_{\text{nearest}} = \arg\min_{v \in V} (v - x)^T S (v - x).$$



Algorithm 4: $\texttt{LQR} - \texttt{RRT}^*((V, E), N)$

1 **for** $i = 1, \ldots, N$ **do**
2 $\quad x_{\text{rand}} \leftarrow \texttt{Sample}$;
3 $\quad x_{\text{nearest}} \leftarrow \texttt{LQRNearest}(V, x_{\text{rand}})$;
4 $\quad x_{\text{new}} \leftarrow \texttt{LQRSteer}(x_{\text{nearest}}, x_{\text{rand}})$;
5 $\quad X_{\text{near}} \leftarrow \texttt{LQRNear}(V, x_{\text{new}})$;
6 $\quad (x_{\min}, \sigma_{\min}) \leftarrow \texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$;
7 $\quad$ **if** $\texttt{CollisionFree}(\sigma)$ **then**
8 $\quad\quad X \leftarrow X \cup \{x_{\text{new}}\}$;
9 $\quad\quad E \leftarrow E \cup \{(x_{\min}, x_{\text{new}})\}$;
10 $\quad\quad (V, E) \leftarrow \texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$;
11 **return** $G = (V, E)$;

Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

1 $\text{minCost} \leftarrow \infty$; $x_{\min} \leftarrow \texttt{NULL}$; $\sigma_{\min} \leftarrow \texttt{NULL}$;
2 **for** $x_{\text{near}} \in X_{\text{near}}$ **do**
3 $\quad \sigma \leftarrow \texttt{LQRSteer}(x_{\text{near}}, x_{\text{new}})$;
4 $\quad$ **if** $\texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma) < \text{minCost}$ **then**
5 $\quad\quad \text{minCost} \leftarrow \texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma)$;
6 $\quad\quad x_{\min} \leftarrow x_{\text{near}}$; $\sigma_{\min} \leftarrow \sigma$;
7 **return** $(x_{\min}, \sigma_{\min})$;

Algorithm 6: $\texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$

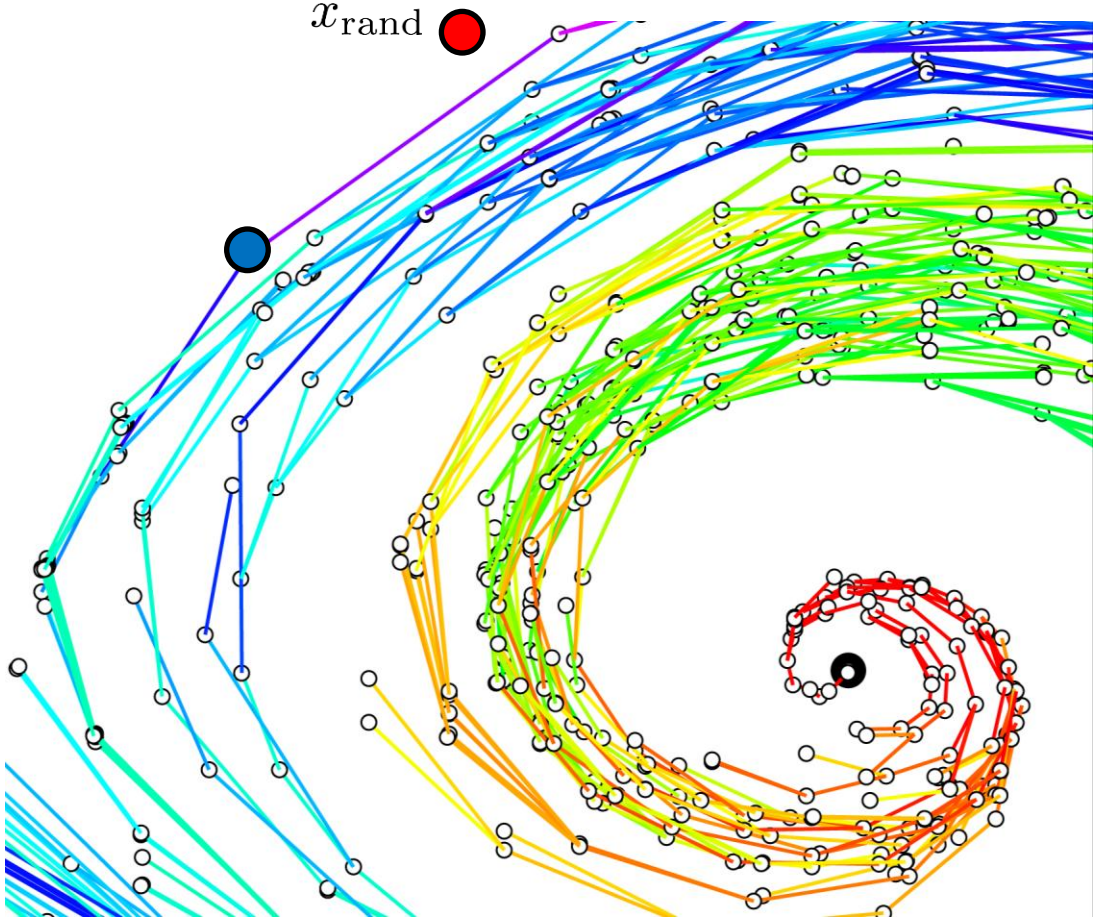
1 **for** $x_{\text{near}} \in X_{\text{near}}$ **do**
2 $\quad \sigma \leftarrow \texttt{LQRSteer}(x_{\text{new}}, x_{\text{near}})$;
3 $\quad$ **if** $\texttt{Cost}(x_{\text{new}}) + \texttt{Cost}(\sigma) < \texttt{Cost}(x_{\text{near}})$ **then**
4 $\quad\quad$ **if** $\texttt{CollisionFree}(\sigma)$ **then**
5 $\quad\quad\quad x_{\text{parent}} \leftarrow \texttt{Parent}(x_{\text{near}})$;
6 $\quad\quad\quad E \leftarrow E \setminus \{x_{\text{parent}}, x_{\text{near}}\}$;
7 $\quad\quad\quad E \leftarrow E \cup \{x_{\text{new}}, x_{\text{near}}\}$;
8 **return** (V, E);

CSAIL

LQR-RRT*

$$[K(x), S(x)] = LQR(A(x,0), B(x,0), Q, R).$$

$$x_{\text{nearest}} = \arg\min_{v \in V} (v - x)^T S (v - x).$$



Algorithm 4: $\texttt{LQR} - \texttt{RRT}^*((V, E), N)$

```
for i = 1, ..., N do
    x_rand ← Sample;
    x_nearest ← LQRNearest(V, x_rand);
    x_new ← LQRSteer(x_nearest, x_rand);
    X_near ← LQRNear(V, x_new);
    (x_min, σ_min) ← ChooseParent(X_near, x_new);
    if CollisionFree(σ) then
        X ← X ∪ {x_new};
        E ← E ∪ {(x_min, x_new)};
        (V, E) ← Rewire( (V, E), X_near, x_new );
return G = (V, E);
```

Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

```
minCost ← ∞;  x_min ← NULL;  σ_min ← NULL;
for x_near ∈ X_near do
    σ ← LQRSteer(x_near, x_new);
    if Cost(x_near) + Cost(σ) < minCost then
        minCost ← Cost(x_near) + Cost(σ);
        x_min ← x_near;  σ_min ← σ;
return (x_min, σ_min);
```

Algorithm 6: $\texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$

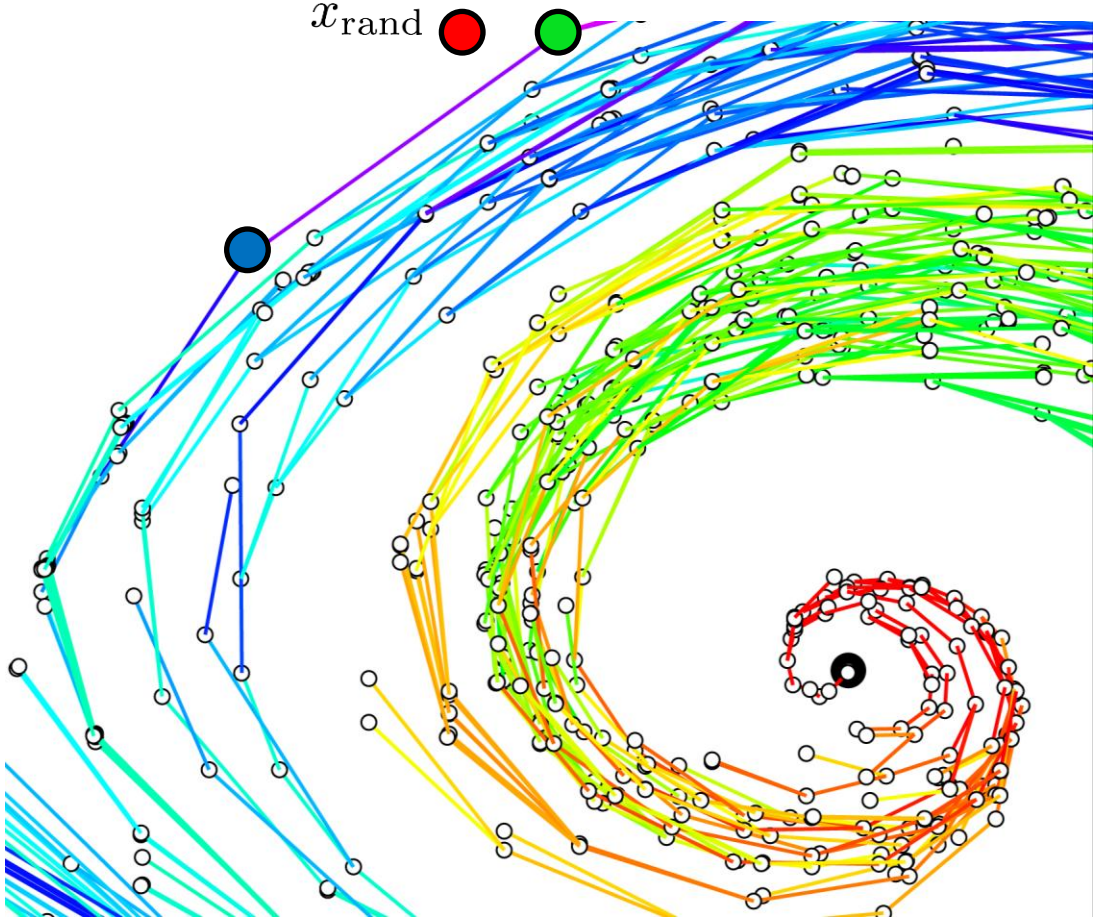
```
for x_near ∈ X_near do
    σ ← LQRSteer(x_new, x_near) ;
    if Cost(x_new) + Cost(σ) < Cost(x_near) then
        if CollisionFree(σ) then
            x_parent ← Parent(x_near);
            E ← E \ {x_parent, x_near};
            E ← E ∪ {x_new, x_near};
return (V, E);
```


CSAIL

LQR-RRT*

$$[K(x), S(x)] = LQR(A(x,0), B(x,0), Q, R).$$

$$x_{\text{nearest}} = \arg\min_{v \in V} (v - x)^T S (v - x).$$



Algorithm 4: $\texttt{LQR} - \texttt{RRT}^*((V, E), N)$

```
for i = 1, ..., N do
    x_rand ← Sample;
    x_nearest ← LQRNearest(V, x_rand);
    x_new ← LQRSteer(x_nearest, x_rand);
    X_near ← LQRNear(V, x_new);
    (x_min, σ_min) ← ChooseParent(X_near, x_new);
    if CollisionFree(σ) then
        X ← X ∪ {x_new};
        E ← E ∪ {(x_min, x_new)};
        (V, E) ← Rewire( (V, E), X_near, x_new );
return G = (V, E);
```

Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

```
minCost ← ∞;  x_min ← NULL;  σ_min ← NULL;
for x_near ∈ X_near do
    σ ← LQRSteer(x_near, x_new);
    if Cost(x_near) + Cost(σ) < minCost then
        minCost ← Cost(x_near) + Cost(σ);
        x_min ← x_near;  σ_min ← σ;
return (x_min, σ_min);
```

Algorithm 6: $\texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$

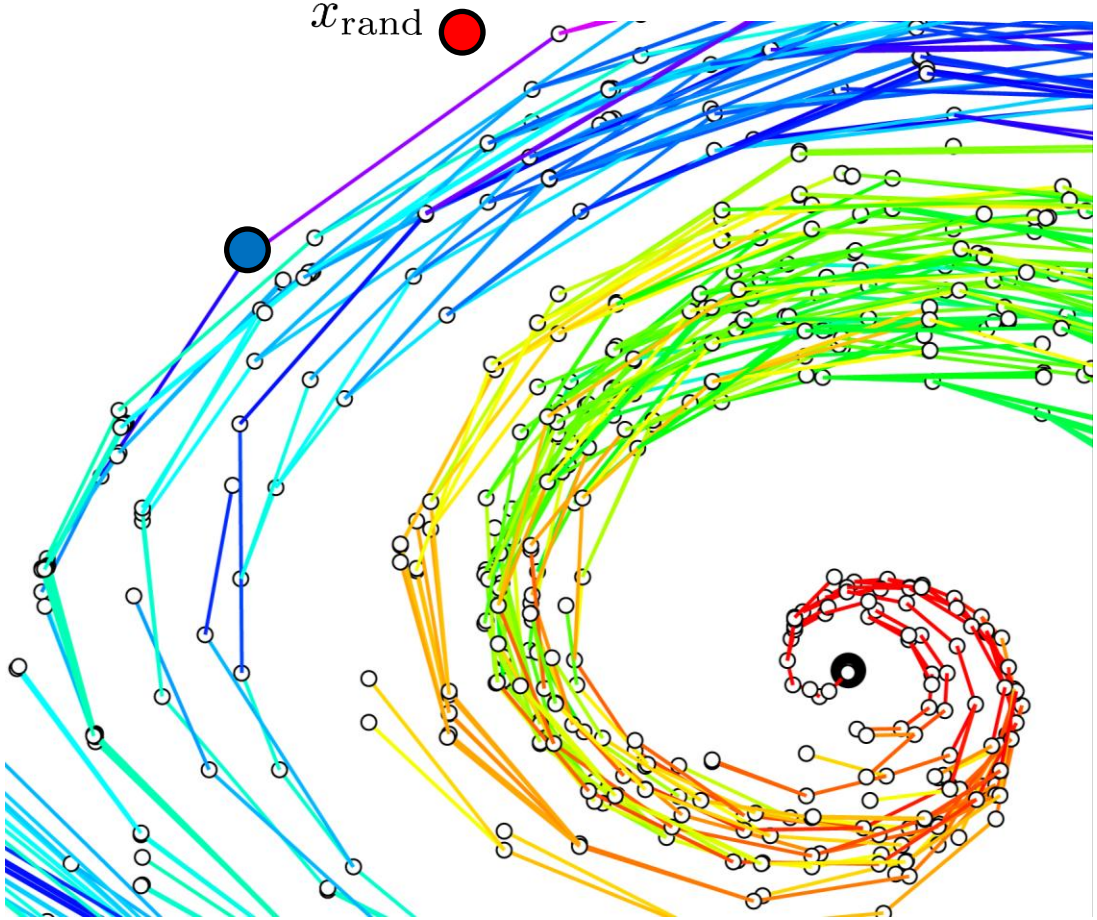
```
for x_near ∈ X_near do
    σ ← LQRSteer(x_new, x_near) ;
    if Cost(x_new) + Cost(σ) < Cost(x_near) then
        if CollisionFree(σ) then
            x_parent ← Parent(x_near);
            E ← E \ {x_parent, x_near};
            E ← E ∪ {x_new, x_near};
return (V, E);
```


CSAIL

LQR-RRT*

$$[K(x), S(x)] = LQR(A(x,0), B(x,0), Q, R).$$

$$x_{\text{nearest}} = \arg\min_{v \in V} (v - x)^T S (v - x).$$



Algorithm 4: $\texttt{LQR} - \texttt{RRT}^*((V, E), N)$

```
for i = 1, ..., N do
    x_rand ← Sample;
    x_nearest ← LQRNearest(V, x_rand);
    x_new ← LQRSteer(x_nearest, x_rand);
    X_near ← LQRNear(V, x_new);
    (x_min, σ_min) ← ChooseParent(X_near, x_new);
    if CollisionFree(σ) then
        X ← X ∪ {x_new};
        E ← E ∪ {(x_min, x_new)};
        (V, E) ← Rewire( (V, E), X_near, x_new );
return G = (V, E);
```

Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

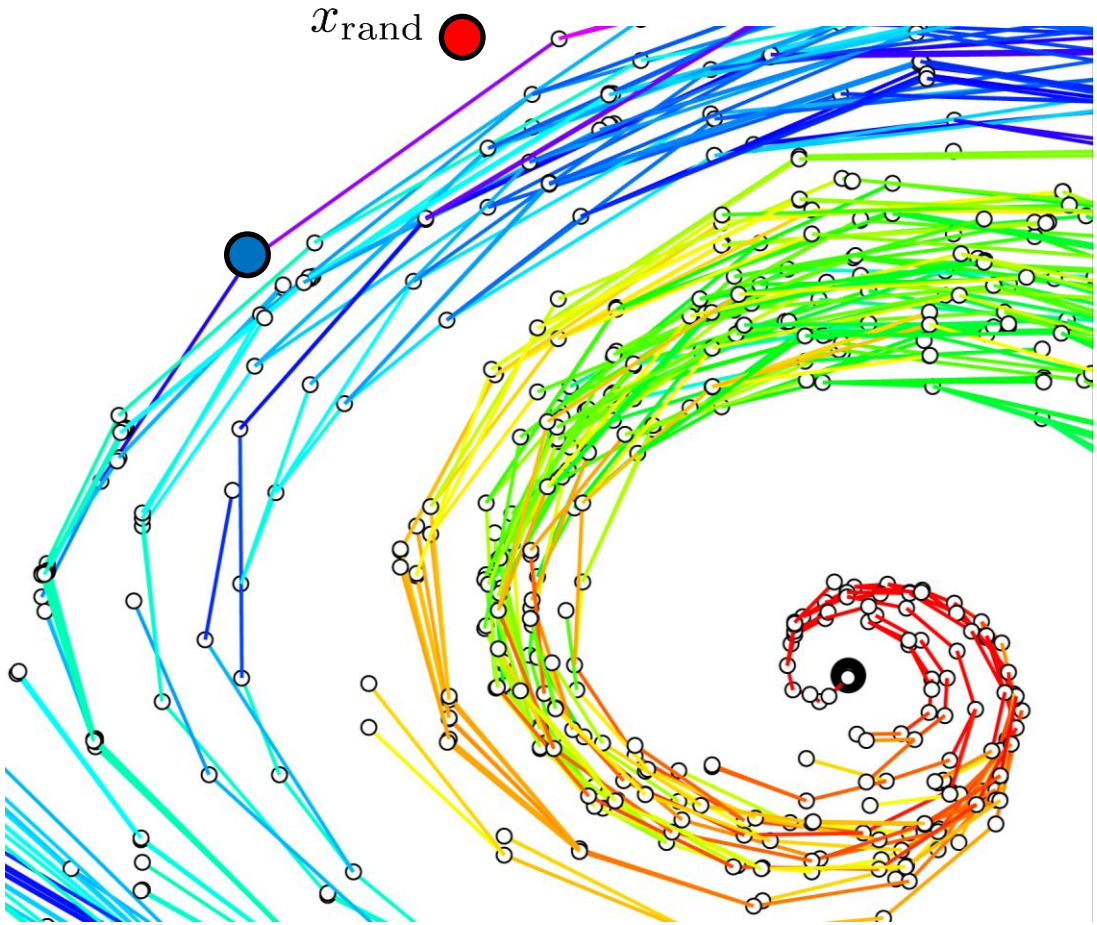
```
minCost ← ∞;  x_min ← NULL;  σ_min ← NULL;
for x_near ∈ X_near do
    σ ← LQRSteer(x_near, x_new);
    if Cost(x_near) + Cost(σ) < minCost then
        minCost ← Cost(x_near) + Cost(σ);
        x_min ← x_near;  σ_min ← σ;
return (x_min, σ_min);
```

Algorithm 6: $\texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$

```
for x_near ∈ X_near do
    σ ← LQRSteer(x_new, x_near) ;
    if Cost(x_new) + Cost(σ) < Cost(x_near) then
        if CollisionFree(σ) then
            x_parent ← Parent(x_near);
            E ← E \ {x_parent, x_near};
            E ← E ∪ {x_new, x_near};
return (V, E);
```


CSAIL

LQR-RRT*



Algorithm 4: $\texttt{LQR} - \texttt{RRT}^*((V, E), N)$

for $i = 1, \ldots, N$ **do**
  $x_{\text{rand}} \leftarrow \texttt{Sample}$;
  $x_{\text{nearest}} \leftarrow \texttt{LQRNearest}(V, x_{\text{rand}})$;
  $x_{\text{new}} \leftarrow \texttt{LQRSteer}(x_{\text{nearest}}, x_{\text{rand}})$;
  $X_{\text{near}} \leftarrow \texttt{LQRNear}(V, x_{\text{new}})$;
  $(x_{\min}, \sigma_{\min}) \leftarrow \texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$;
  if $\texttt{CollisionFree}(\sigma)$ **then**
    $X \leftarrow X \cup \{x_{\text{new}}\}$;
    $E \leftarrow E \cup \{(x_{\min}, x_{\text{new}})\}$;
    $(V, E) \leftarrow \texttt{Rewire}(\,(V, E), X_{\text{near}}, x_{\text{new}}\,)$;
return $G = (V, E)$;

Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

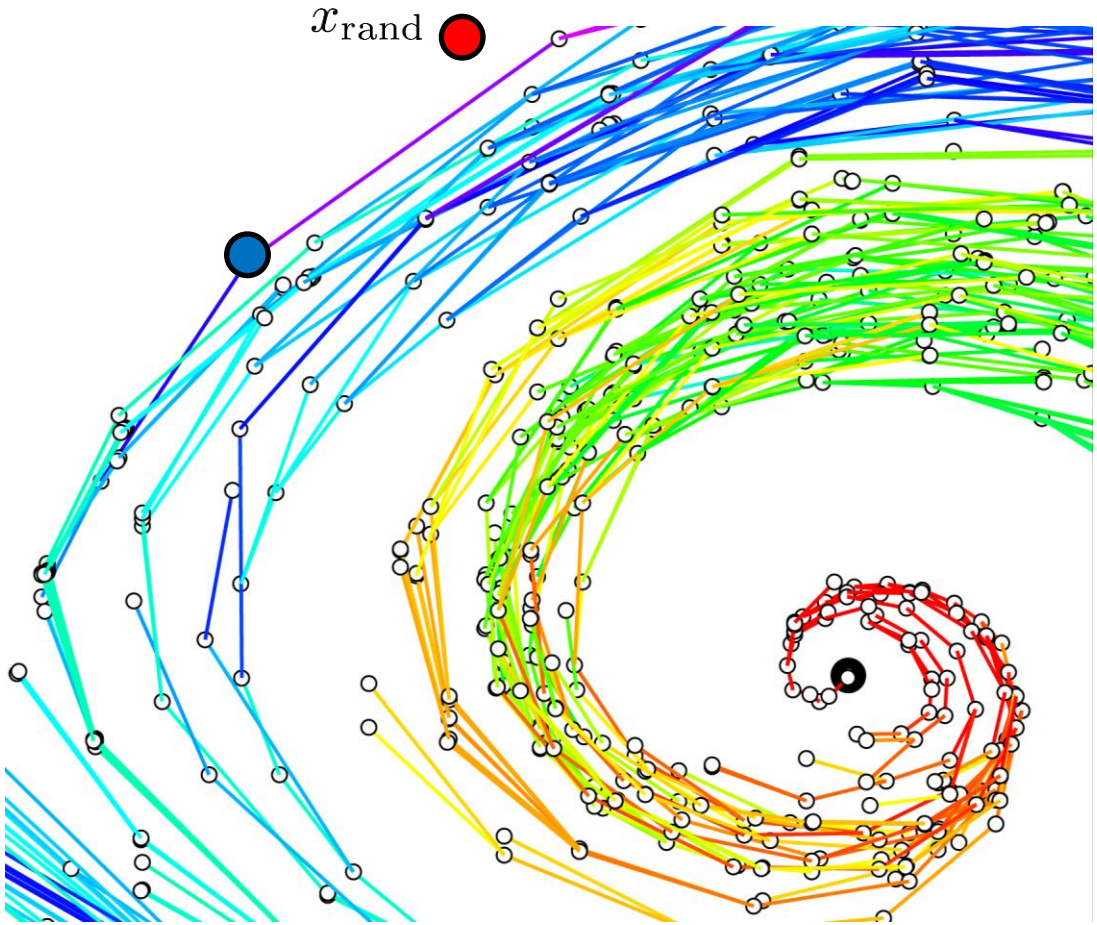
$\text{minCost} \leftarrow \infty$; $x_{\min} \leftarrow \texttt{NULL}$; $\sigma_{\min} \leftarrow \texttt{NULL}$;
for $x_{\text{near}} \in X_{\text{near}}$ **do**
  $\sigma \leftarrow \texttt{LQRSteer}(x_{\text{near}}, x_{\text{new}})$;
  if $\texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma) < \text{minCost}$ **then**
    $\text{minCost} \leftarrow \texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma)$;
    $x_{\min} \leftarrow x_{\text{near}}$; $\sigma_{\min} \leftarrow \sigma$;
return $(x_{\min}, \sigma_{\min})$;

Algorithm 6: $\texttt{Rewire}(\,(V, E), X_{\text{near}}, x_{\text{new}}\,)$

for $x_{\text{near}} \in X_{\text{near}}$ **do**
  $\sigma \leftarrow \texttt{LQRSteer}(x_{\text{new}}, x_{\text{near}})$;
  if $\texttt{Cost}(x_{\text{new}}) + \texttt{Cost}(\sigma) < \texttt{Cost}(x_{\text{near}})$ **then**
    if $\texttt{CollisionFree}(\sigma)$ **then**
      $x_{\text{parent}} \leftarrow \texttt{Parent}(x_{\text{near}})$;
      $E \leftarrow E \setminus \{x_{\text{parent}}, x_{\text{near}}\}$;
      $E \leftarrow E \cup \{x_{\text{new}}, x_{\text{near}}\}$;
return (V, E);

CSAIL

LQR-RRT*



Algorithm 4: $\texttt{LQR} - \texttt{RRT}^*((V, E), N)$

```
for i = 1, ..., N do
    x_rand ← Sample;
    x_nearest ← LQRNearest(V, x_rand);
    x_new ← LQRSteer(x_nearest, x_rand);
    X_near ← LQRNear(V, x_new);
    (x_min, σ_min) ← ChooseParent(X_near, x_new);
    if CollisionFree(σ) then
        X ← X ∪ {x_new};
        E ← E ∪ {(x_min, x_new)};
        (V, E) ← Rewire( (V, E), X_near, x_new );
return G = (V, E);
```

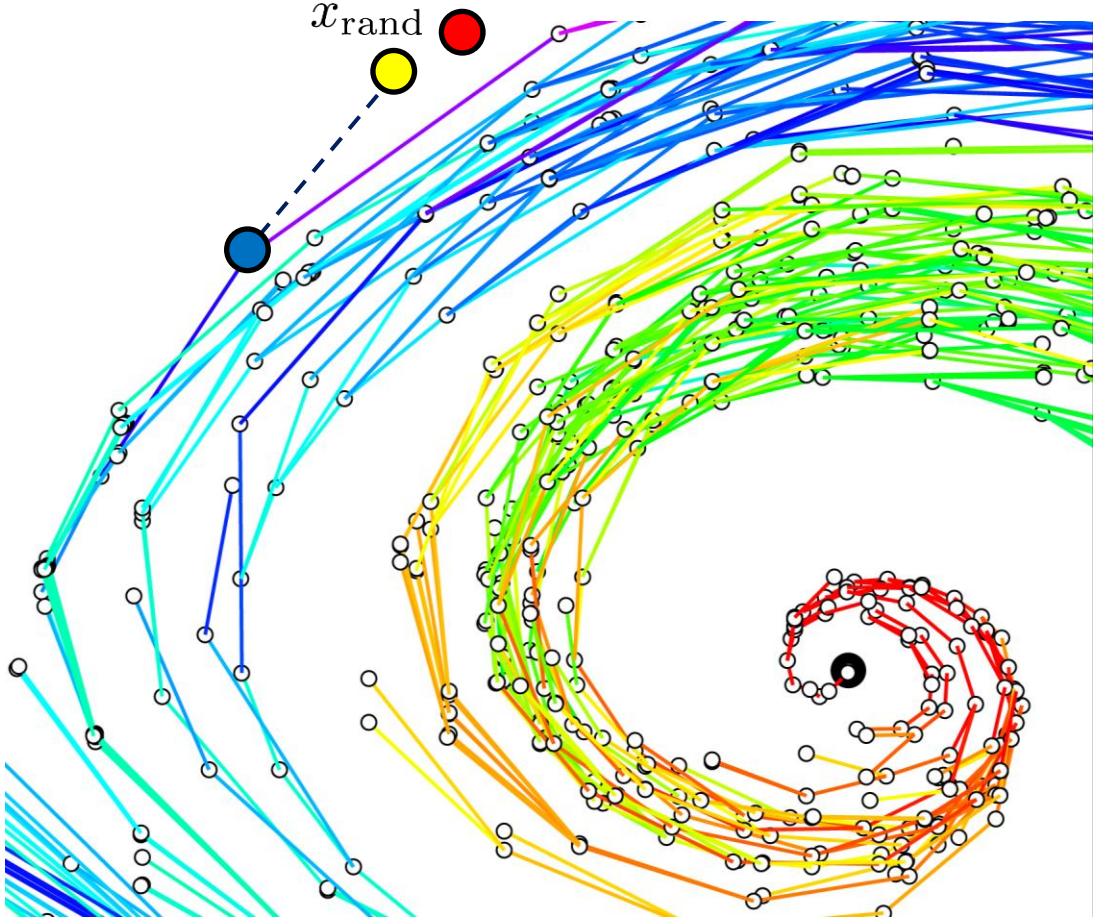
Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

```
minCost ← ∞;  x_min ← NULL;  σ_min ← NULL;
for x_near ∈ X_near do
    σ ← LQRSteer(x_near, x_new);
    if Cost(x_near) + Cost(σ) < minCost then
        minCost ← Cost(x_near) + Cost(σ);
        x_min ← x_near;  σ_min ← σ;
return (x_min, σ_min);
```

Algorithm 6: $\texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$

```
for x_near ∈ X_near do
    σ ← LQRSteer(x_new, x_near) ;
    if Cost(x_new) + Cost(σ) < Cost(x_near) then
        if CollisionFree(σ) then
            x_parent ← Parent(x_near);
            E ← E \ {x_parent, x_near};
            E ← E ∪ {x_new, x_near};
return (V, E);
```

LQR-RRT*



Algorithm 4: $\texttt{LQR} - \texttt{RRT}^*((V, E), N)$

```
for i = 1, ..., N do
    x_rand ← Sample;
    x_nearest ← LQRNearest(V, x_rand);
    x_new ← LQRSteer(x_nearest, x_rand);
    X_near ← LQRNear(V, x_new);
    (x_min, σ_min) ← ChooseParent(X_near, x_new);
    if CollisionFree(σ) then
        X ← X ∪ {x_new};
        E ← E ∪ {(x_min, x_new)};
        (V, E) ← Rewire( (V, E), X_near, x_new );
return G = (V, E);
```

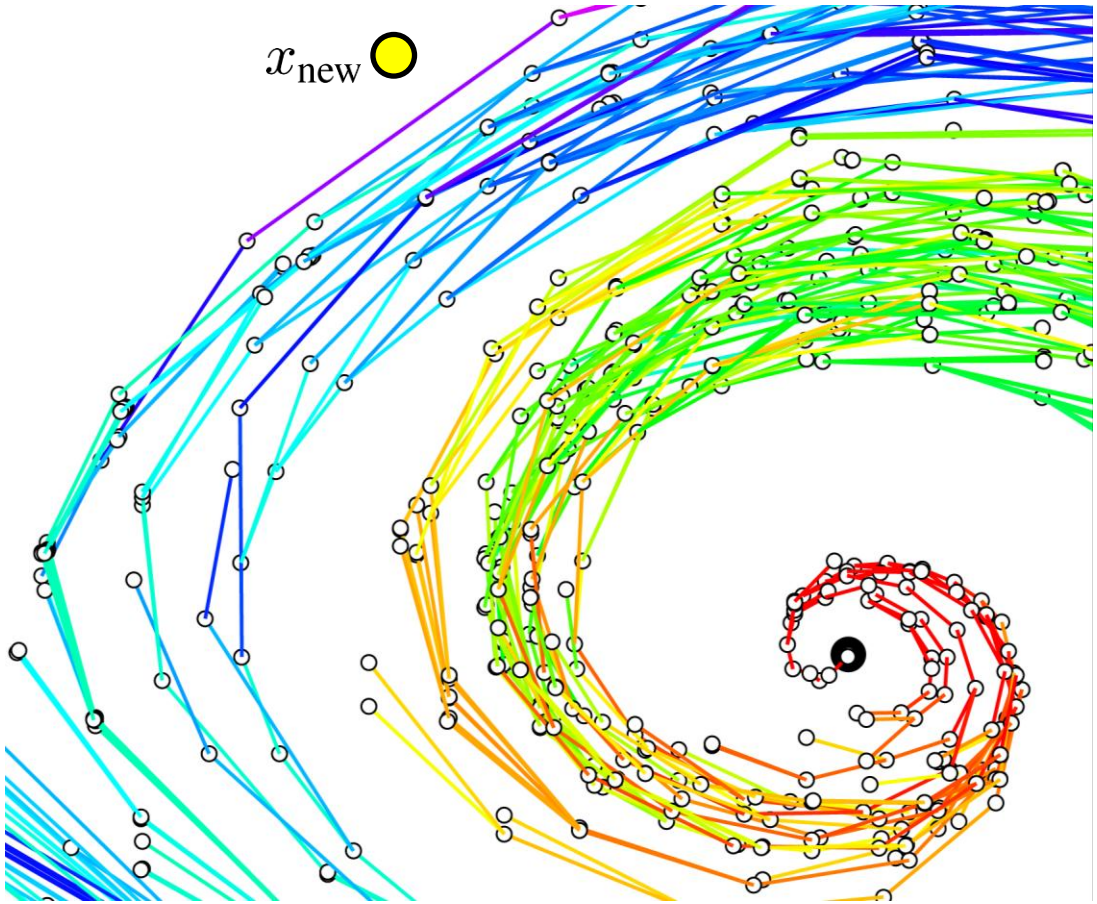
Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

```
minCost ← ∞;  x_min ← NULL;  σ_min ← NULL;
for x_near ∈ X_near do
    σ ← LQRSteer(x_near, x_new);
    if Cost(x_near) + Cost(σ) < minCost then
        minCost ← Cost(x_near) + Cost(σ);
        x_min ← x_near;  σ_min ← σ;
return (x_min, σ_min);
```

Algorithm 6: $\texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$

```
for x_near ∈ X_near do
    σ ← LQRSteer(x_new, x_near) ;
    if Cost(x_new) + Cost(σ) < Cost(x_near) then
        if CollisionFree(σ) then
            x_parent ← Parent(x_near);
            E ← E \ {x_parent, x_near};
            E ← E ∪ {x_new, x_near};
return (V, E);
```

LQR-RRT*



Algorithm 4: $\texttt{LQR}-\texttt{RRT}^*((V,E),N)$

```
for i = 1, ..., N do
    x_rand ← Sample;
    x_nearest ← LQRNearest(V, x_rand);
    x_new ← LQRSteer(x_nearest, x_rand);
    X_near ← LQRNear(V, x_new);
    (x_min, σ_min) ← ChooseParent(X_near, x_new);
    if CollisionFree(σ) then
        X ← X ∪ {x_new};
        E ← E ∪ {(x_min, x_new)};
        (V, E) ← Rewire( (V, E), X_near, x_new );
return G = (V, E);
```

Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

```
minCost ← ∞;  x_min ← NULL;  σ_min ← NULL;
for x_near ∈ X_near do
    σ ← LQRSteer(x_near, x_new);
    if Cost(x_near) + Cost(σ) < minCost then
        minCost ← Cost(x_near) + Cost(σ);
        x_min ← x_near;  σ_min ← σ;
return (x_min, σ_min);
```

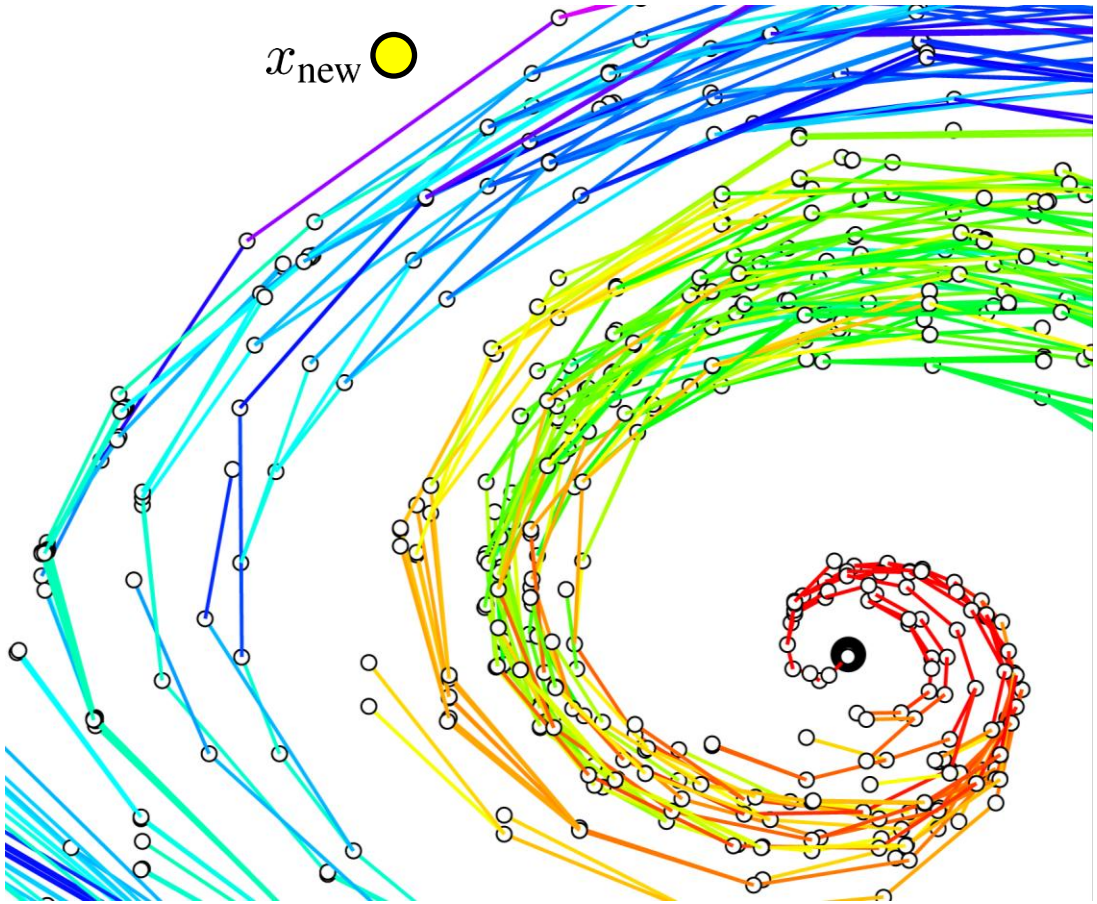
Algorithm 6: $\texttt{Rewire}(\,(V,E), X_{\text{near}}, x_{\text{new}}\,)$

```
for x_near ∈ X_near do
    σ ← LQRSteer(x_new, x_near) ;
    if Cost(x_new) + Cost(σ) < Cost(x_near) then
        if CollisionFree(σ) then
            x_parent ← Parent(x_near);
            E ← E \ {x_parent, x_near};
            E ← E ∪ {x_new, x_near};
return (V, E);
```


CSAIL

LQR-RRT*

$$\texttt{LQRNear}(V, x)$$
$$= \left\{ v \in V : (v-x)^T S (v-x) \leq \gamma \left(\frac{\log n}{n} \right)^{\frac{1}{d}} \right\}$$



Algorithm 4: $\texttt{LQR-RRT}^*((V, E), N)$

```
for i = 1, ..., N do
    x_rand ← Sample;
    x_nearest ← LQRNearest(V, x_rand);
    x_new ← LQRSteer(x_nearest, x_rand);
    X_near ← LQRNear(V, x_new);
    (x_min, σ_min) ← ChooseParent(X_near, x_new);
    if CollisionFree(σ) then
        X ← X ∪ {x_new};
        E ← E ∪ {(x_min, x_new)};
        (V, E) ← Rewire( (V, E), X_near, x_new );
return G = (V, E);
```

Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

```
minCost ← ∞; x_min ← NULL; σ_min ← NULL;
for x_near ∈ X_near do
    σ ← LQRSteer(x_near, x_new);
    if Cost(x_near) + Cost(σ) < minCost then
        minCost ← Cost(x_near) + Cost(σ);
        x_min ← x_near; σ_min ← σ;
return (x_min, σ_min);
```

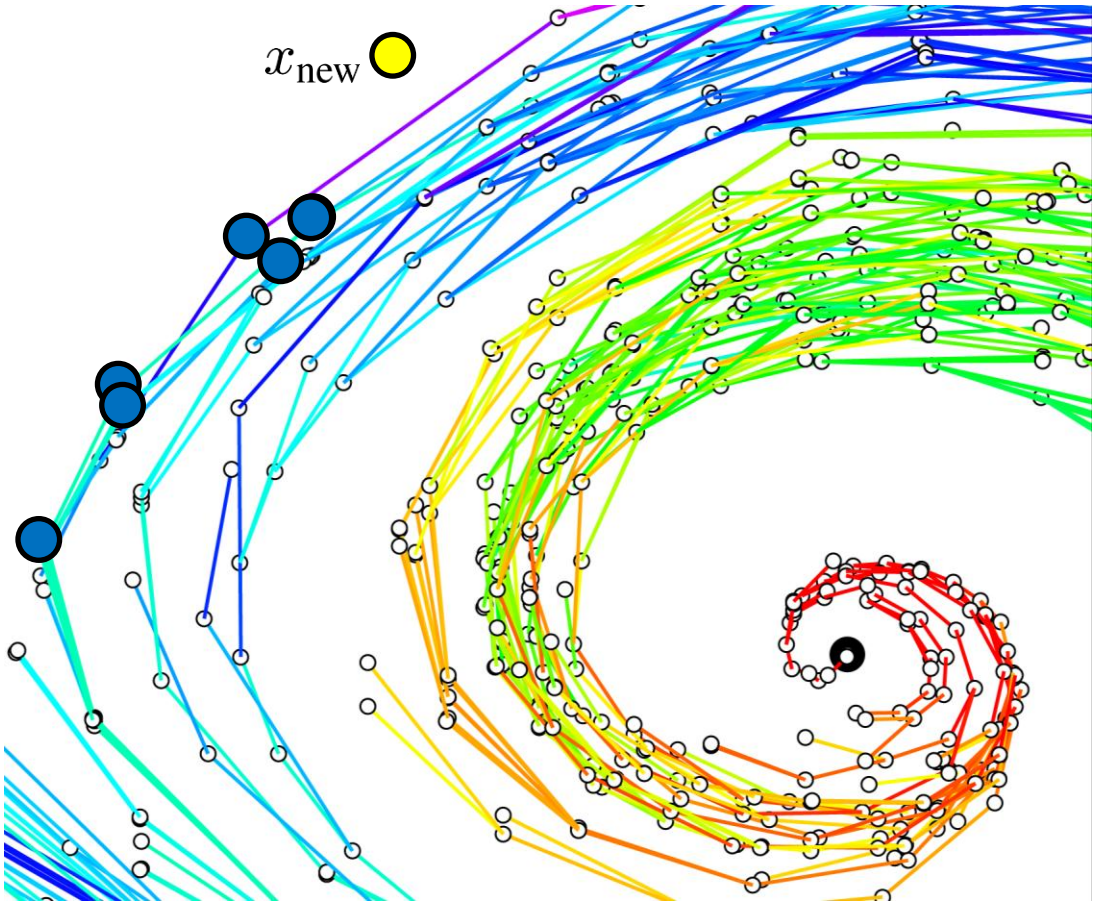
Algorithm 6: $\texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$

```
for x_near ∈ X_near do
    σ ← LQRSteer(x_new, x_near) ;
    if Cost(x_new) + Cost(σ) < Cost(x_near) then
        if CollisionFree(σ) then
            x_parent ← Parent(x_near);
            E ← E \ {x_parent, x_near};
            E ← E ∪ {x_new, x_near};
return (V, E);
```


CSAIL

LQR-RRT*

$$\texttt{LQRNear}(V, x)$$
$$= \left\{ v \in V : (v-x)^T S (v-x) \leq \gamma \left(\frac{\log n}{n} \right)^{\frac{1}{d}} \right\}$$



Algorithm 4: $\texttt{LQR} - \texttt{RRT}^*((V, E), N)$

1 **for** $i = 1, \ldots, N$ **do**
2 $x_{\text{rand}} \leftarrow \texttt{Sample}$;
3 $x_{\text{nearest}} \leftarrow \texttt{LQRNearest}(V, x_{\text{rand}})$;
4 $x_{\text{new}} \leftarrow \texttt{LQRSteer}(x_{\text{nearest}}, x_{\text{rand}})$;
5 $X_{\text{near}} \leftarrow \texttt{LQRNear}(V, x_{\text{new}})$;
6 $(x_{\min}, \sigma_{\min}) \leftarrow \texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$;
7 **if** $\texttt{CollisionFree}(\sigma)$ **then**
8 $X \leftarrow X \cup \{x_{\text{new}}\}$;
9 $E \leftarrow E \cup \{(x_{\min}, x_{\text{new}})\}$;
10 $(V, E) \leftarrow \texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$;
11 **return** $G = (V, E)$;

Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

1 $\text{minCost} \leftarrow \infty$; $x_{\min} \leftarrow \texttt{NULL}$; $\sigma_{\min} \leftarrow \texttt{NULL}$;
2 **for** $x_{\text{near}} \in X_{\text{near}}$ **do**
3 $\sigma \leftarrow \texttt{LQRSteer}(x_{\text{near}}, x_{\text{new}})$;
4 **if** $\texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma) < \text{minCost}$ **then**
5 $\text{minCost} \leftarrow \texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma)$;
6 $x_{\min} \leftarrow x_{\text{near}}$; $\sigma_{\min} \leftarrow \sigma$;
7 **return** $(x_{\min}, \sigma_{\min})$;

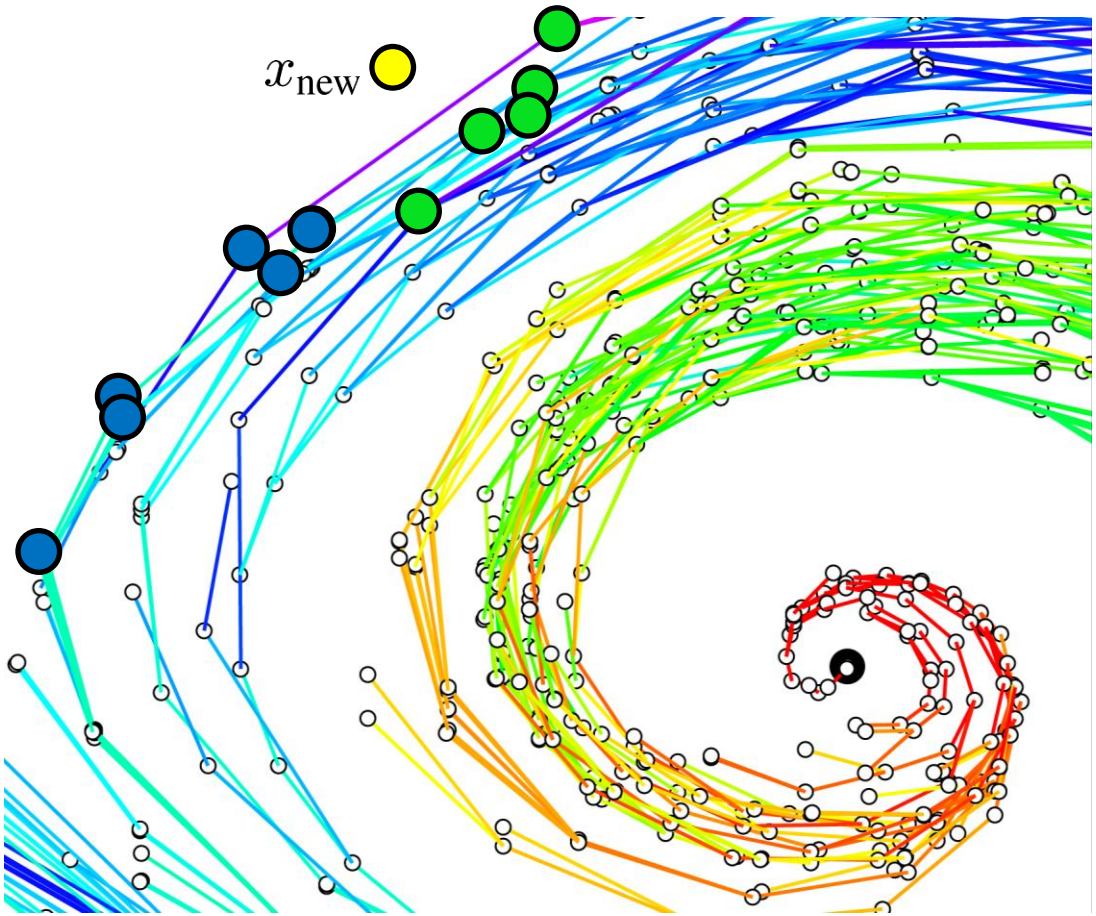
Algorithm 6: $\texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$

1 **for** $x_{\text{near}} \in X_{\text{near}}$ **do**
2 $\sigma \leftarrow \texttt{LQRSteer}(x_{\text{new}}, x_{\text{near}})$;
3 **if** $\texttt{Cost}(x_{\text{new}}) + \texttt{Cost}(\sigma) < \texttt{Cost}(x_{\text{near}})$ **then**
4 **if** $\texttt{CollisionFree}(\sigma)$ **then**
5 $x_{\text{parent}} \leftarrow \texttt{Parent}(x_{\text{near}})$;
6 $E \leftarrow E \setminus \{x_{\text{parent}}, x_{\text{near}}\}$;
7 $E \leftarrow E \cup \{x_{\text{new}}, x_{\text{near}}\}$;
8 **return** (V, E);

CSAIL

LQR-RRT*

$$\texttt{LQRNear}(V, x) = \left\{ v \in V : (v-x)^T S (v-x) \leq \gamma \left(\frac{\log n}{n} \right)^{\frac{1}{d}} \right\}$$



Algorithm 4: $\texttt{LQR} - \texttt{RRT}^*((V, E), N)$

```
for i = 1, ..., N do
    x_rand ← Sample;
    x_nearest ← LQRNearest(V, x_rand);
    x_new ← LQRSteer(x_nearest, x_rand);
    X_near ← LQRNear(V, x_new);
    (x_min, σ_min) ← ChooseParent(X_near, x_new);
    if CollisionFree(σ) then
        X ← X ∪ {x_new};
        E ← E ∪ {(x_min, x_new)};
        (V, E) ← Rewire( (V, E), X_near, x_new );
return G = (V, E);
```

Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

```
minCost ← ∞;  x_min ← NULL;  σ_min ← NULL;
for x_near ∈ X_near do
    σ ← LQRSteer(x_near, x_new);
    if Cost(x_near) + Cost(σ) < minCost then
        minCost ← Cost(x_near) + Cost(σ);
        x_min ← x_near;  σ_min ← σ;
return (x_min, σ_min);
```

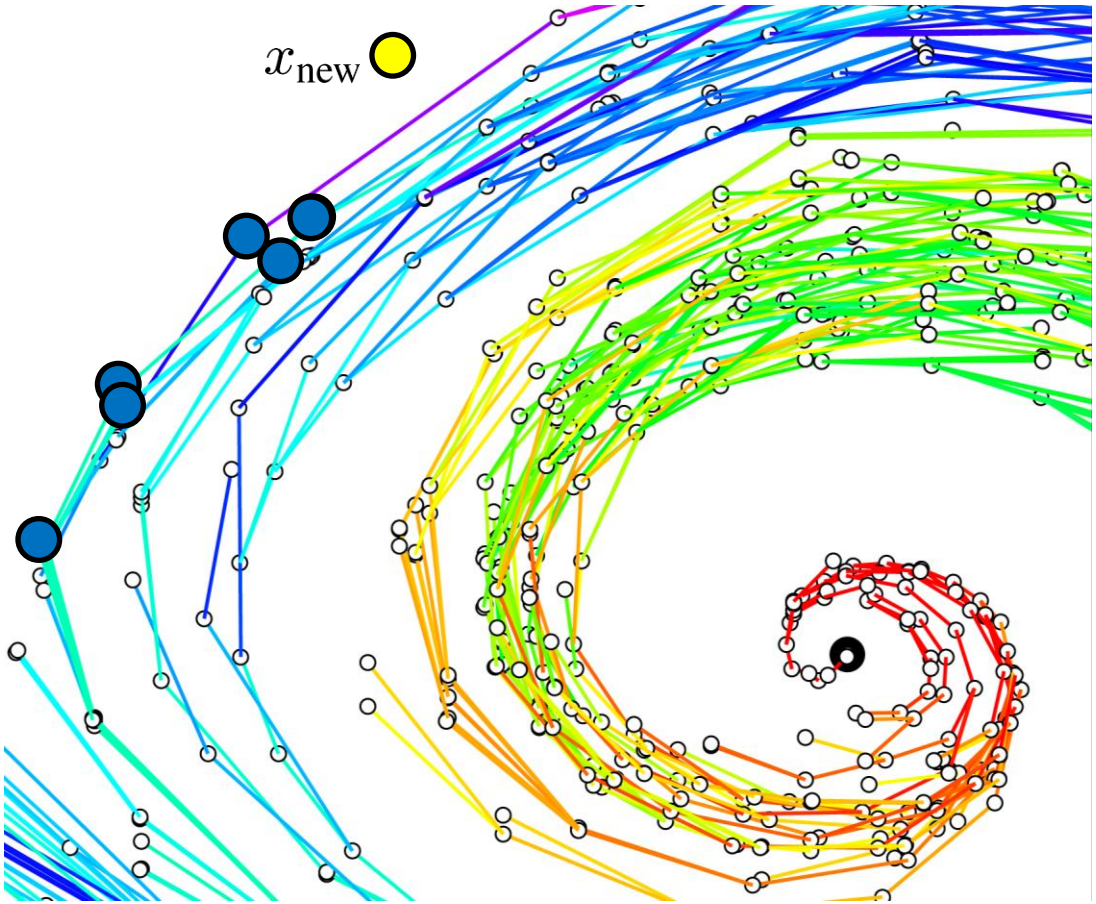
Algorithm 6: $\texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$

```
for x_near ∈ X_near do
    σ ← LQRSteer(x_new, x_near) ;
    if Cost(x_new) + Cost(σ) < Cost(x_near) then
        if CollisionFree(σ) then
            x_parent ← Parent(x_near);
            E ← E \ {x_parent, x_near};
            E ← E ∪ {x_new, x_near};
return (V, E);
```


CSAIL

LQR-RRT*

$$\texttt{LQRNear}(V, x)$$
$$= \left\{ v \in V : (v-x)^T S (v-x) \leq \gamma \left(\frac{\log n}{n} \right)^{\frac{1}{d}} \right\}$$



Algorithm 4: $\texttt{LQR} - \texttt{RRT}^*((V, E), N)$

1 **for** $i = 1, \ldots, N$ **do**
2 $x_{\text{rand}} \leftarrow \texttt{Sample}$;
3 $x_{\text{nearest}} \leftarrow \texttt{LQRNearest}(V, x_{\text{rand}})$;
4 $x_{\text{new}} \leftarrow \texttt{LQRSteer}(x_{\text{nearest}}, x_{\text{rand}})$;
5 $X_{\text{near}} \leftarrow \texttt{LQRNear}(V, x_{\text{new}})$;
6 $(x_{\min}, \sigma_{\min}) \leftarrow \texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$;
7 **if** $\texttt{CollisionFree}(\sigma)$ **then**
8 $X \leftarrow X \cup \{x_{\text{new}}\}$;
9 $E \leftarrow E \cup \{(x_{\min}, x_{\text{new}})\}$;
10 $(V, E) \leftarrow \texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$;
11 **return** $G = (V, E)$;

Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

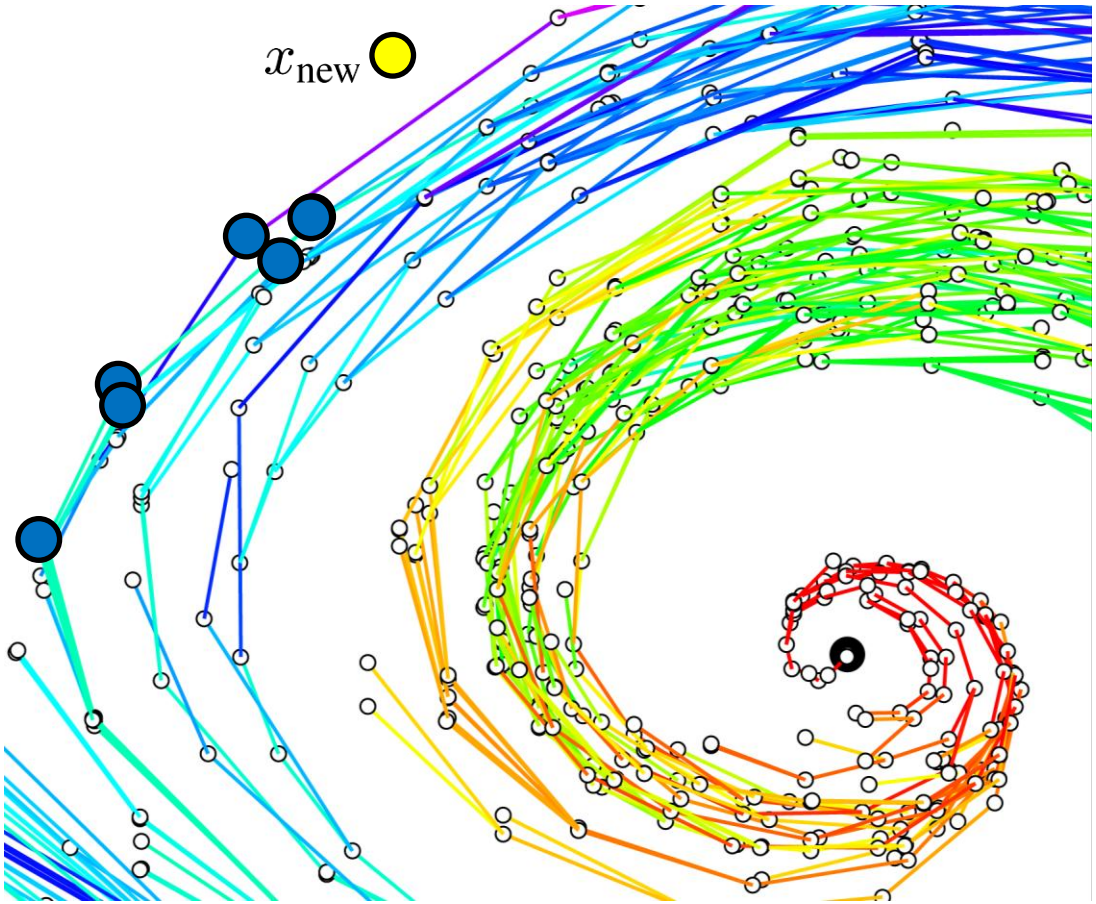
1 $\text{minCost} \leftarrow \infty$; $x_{\min} \leftarrow \texttt{NULL}$; $\sigma_{\min} \leftarrow \texttt{NULL}$;
2 **for** $x_{\text{near}} \in X_{\text{near}}$ **do**
3 $\sigma \leftarrow \texttt{LQRSteer}(x_{\text{near}}, x_{\text{new}})$;
4 **if** $\texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma) < \text{minCost}$ **then**
5 $\text{minCost} \leftarrow \texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma)$;
6 $x_{\min} \leftarrow x_{\text{near}}$; $\sigma_{\min} \leftarrow \sigma$;
7 **return** $(x_{\min}, \sigma_{\min})$;

Algorithm 6: $\texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$

1 **for** $x_{\text{near}} \in X_{\text{near}}$ **do**
2 $\sigma \leftarrow \texttt{LQRSteer}(x_{\text{new}}, x_{\text{near}})$;
3 **if** $\texttt{Cost}(x_{\text{new}}) + \texttt{Cost}(\sigma) < \texttt{Cost}(x_{\text{near}})$ **then**
4 **if** $\texttt{CollisionFree}(\sigma)$ **then**
5 $x_{\text{parent}} \leftarrow \texttt{Parent}(x_{\text{near}})$;
6 $E \leftarrow E \setminus \{x_{\text{parent}}, x_{\text{near}}\}$;
7 $E \leftarrow E \cup \{x_{\text{new}}, x_{\text{near}}\}$;
8 **return** (V, E);

CSAIL

LQR-RRT*



Algorithm 4: $\texttt{LQR}-\texttt{RRT}^*((V,E),N)$

for $i = 1,\ldots,N$ **do**
  $x_{\text{rand}} \leftarrow \texttt{Sample}$;
  $x_{\text{nearest}} \leftarrow \texttt{LQRNearest}(V, x_{\text{rand}})$;
  $x_{\text{new}} \leftarrow \texttt{LQRSteer}(x_{\text{nearest}}, x_{\text{rand}})$;
  $X_{\text{near}} \leftarrow \texttt{LQRNear}(V, x_{\text{new}})$;
  $(x_{\min}, \sigma_{\min}) \leftarrow \texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$;
  if $\texttt{CollisionFree}(\sigma)$ **then**
    $X \leftarrow X \cup \{x_{\text{new}}\}$;
    $E \leftarrow E \cup \{(x_{\min}, x_{\text{new}})\}$;
    $(V,E) \leftarrow \texttt{Rewire}((V,E), X_{\text{near}}, x_{\text{new}})$;
return $G = (V,E)$;

Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

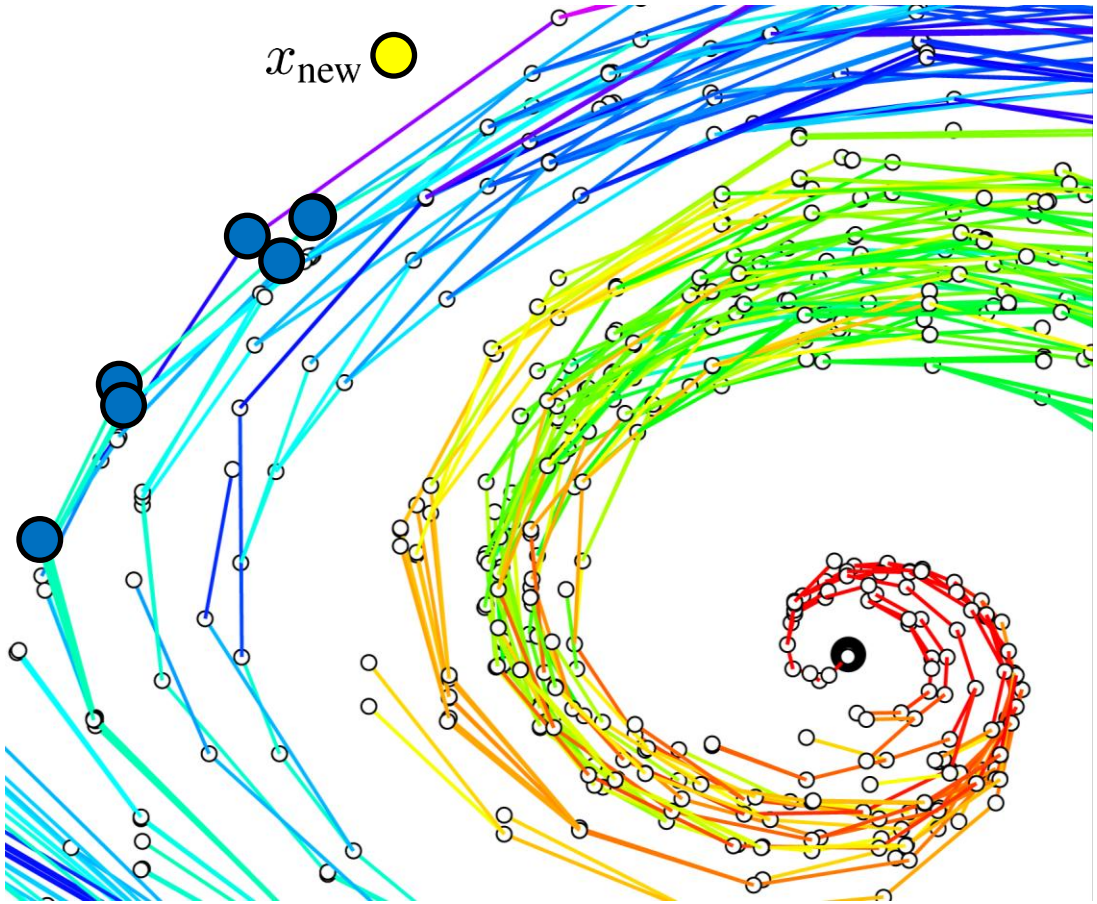
$\text{minCost} \leftarrow \infty$; $x_{\min} \leftarrow \texttt{NULL}$; $\sigma_{\min} \leftarrow \texttt{NULL}$;
for $x_{\text{near}} \in X_{\text{near}}$ **do**
  $\sigma \leftarrow \texttt{LQRSteer}(x_{\text{near}}, x_{\text{new}})$;
  if $\texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma) < \text{minCost}$ **then**
    $\text{minCost} \leftarrow \texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma)$;
    $x_{\min} \leftarrow x_{\text{near}}$; $\sigma_{\min} \leftarrow \sigma$;
return $(x_{\min}, \sigma_{\min})$;

Algorithm 6: $\texttt{Rewire}((V,E), X_{\text{near}}, x_{\text{new}})$

for $x_{\text{near}} \in X_{\text{near}}$ **do**
  $\sigma \leftarrow \texttt{LQRSteer}(x_{\text{new}}, x_{\text{near}})$;
  if $\texttt{Cost}(x_{\text{new}}) + \texttt{Cost}(\sigma) < \texttt{Cost}(x_{\text{near}})$ **then**
    if $\texttt{CollisionFree}(\sigma)$ **then**
      $x_{\text{parent}} \leftarrow \texttt{Parent}(x_{\text{near}})$;
      $E \leftarrow E \setminus \{x_{\text{parent}}, x_{\text{near}}\}$;
      $E \leftarrow E \cup \{x_{\text{new}}, x_{\text{near}}\}$;
return (V,E);

CSAIL

LQR-RRT*



Algorithm 4: $\texttt{LQR}-\texttt{RRT}^*((V,E),N)$

```
for i = 1, ..., N do
    x_rand ← Sample;
    x_nearest ← LQRNearest(V, x_rand);
    x_new ← LQRSteer(x_nearest, x_rand);
    X_near ← LQRNear(V, x_new);
    (x_min, σ_min) ← ChooseParent(X_near, x_new);
    if CollisionFree(σ) then
        X ← X ∪ {x_new};
        E ← E ∪ {(x_min, x_new)};
        (V, E) ← Rewire( (V, E), X_near, x_new );
return G = (V, E);
```

Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

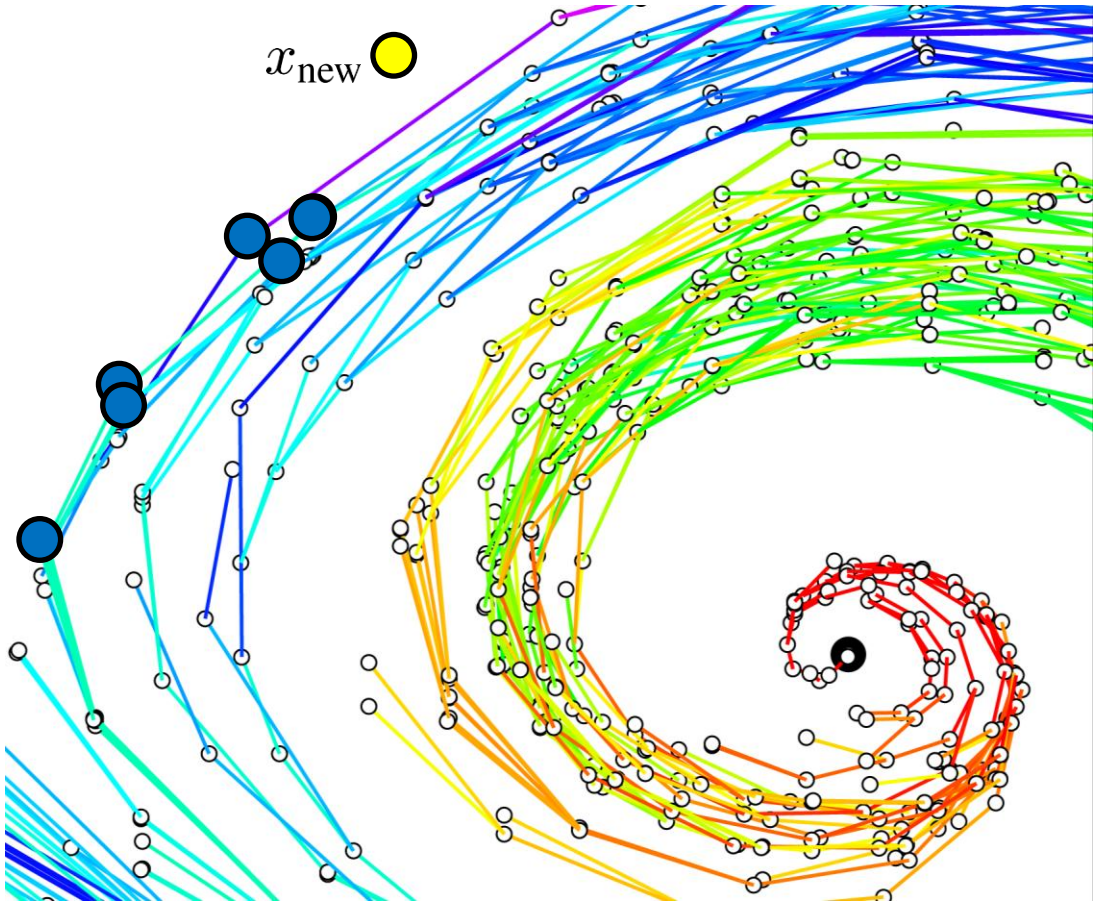
```
minCost ← ∞;  x_min ← NULL;  σ_min ← NULL;
for x_near ∈ X_near do
    σ ← LQRSteer(x_near, x_new);
    if Cost(x_near) + Cost(σ) < minCost then
        minCost ← Cost(x_near) + Cost(σ);
        x_min ← x_near;  σ_min ← σ;
return (x_min, σ_min);
```

Algorithm 6: $\texttt{Rewire}((V,E), X_{\text{near}}, x_{\text{new}})$

```
for x_near ∈ X_near do
    σ ← LQRSteer(x_new, x_near) ;
    if Cost(x_new) + Cost(σ) < Cost(x_near) then
        if CollisionFree(σ) then
            x_parent ← Parent(x_near);
            E ← E \ {x_parent, x_near};
            E ← E ∪ {x_new, x_near};
return (V, E);
```


CSAIL

LQR-RRT*



Algorithm 4: $\texttt{LQR}-\texttt{RRT}^*((V,E),N)$

```
for i = 1, ..., N do
    x_rand ← Sample;
    x_nearest ← LQRNearest(V, x_rand);
    x_new ← LQRSteer(x_nearest, x_rand);
    X_near ← LQRNear(V, x_new);
    (x_min, σ_min) ← ChooseParent(X_near, x_new);
    if CollisionFree(σ) then
        X ← X ∪ {x_new};
        E ← E ∪ {(x_min, x_new)};
        (V, E) ← Rewire( (V, E), X_near, x_new );
return G = (V, E);
```

Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

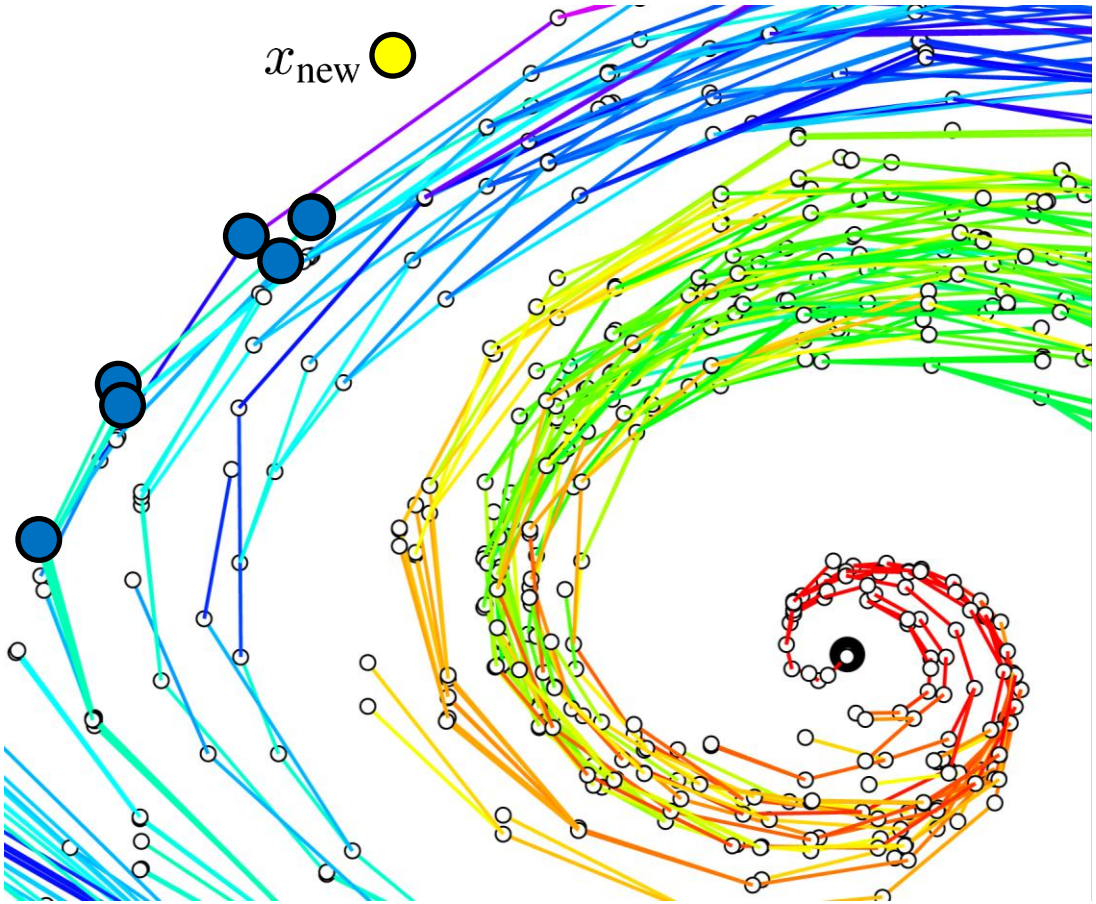
```
minCost ← ∞;  x_min ← NULL;  σ_min ← NULL;
for x_near ∈ X_near do
    σ ← LQRSteer(x_near, x_new);
    if Cost(x_near) + Cost(σ) < minCost then
        minCost ← Cost(x_near) + Cost(σ);
        x_min ← x_near;  σ_min ← σ;
return (x_min, σ_min);
```

Algorithm 6: $\texttt{Rewire}((V,E), X_{\text{near}}, x_{\text{new}})$

```
for x_near ∈ X_near do
    σ ← LQRSteer(x_new, x_near) ;
    if Cost(x_new) + Cost(σ) < Cost(x_near) then
        if CollisionFree(σ) then
            x_parent ← Parent(x_near);
            E ← E \ {x_parent, x_near};
            E ← E ∪ {x_new, x_near};
return (V, E);
```


CSAIL

LQR-RRT*



Algorithm 4: $\texttt{LQR}-\texttt{RRT}^*((V,E),N)$

for $i = 1,\ldots,N$ **do**
  $x_{\text{rand}} \leftarrow \texttt{Sample}$;
  $x_{\text{nearest}} \leftarrow \texttt{LQRNearest}(V, x_{\text{rand}})$;
  $x_{\text{new}} \leftarrow \texttt{LQRSteer}(x_{\text{nearest}}, x_{\text{rand}})$;
  $X_{\text{near}} \leftarrow \texttt{LQRNear}(V, x_{\text{new}})$;
  $(x_{\min}, \sigma_{\min}) \leftarrow \texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$;
  if $\texttt{CollisionFree}(\sigma)$ **then**
    $X \leftarrow X \cup \{x_{\text{new}}\}$;
    $E \leftarrow E \cup \{(x_{\min}, x_{\text{new}})\}$;
    $(V,E) \leftarrow \texttt{Rewire}((V,E), X_{\text{near}}, x_{\text{new}})$;
return $G = (V,E)$;

Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

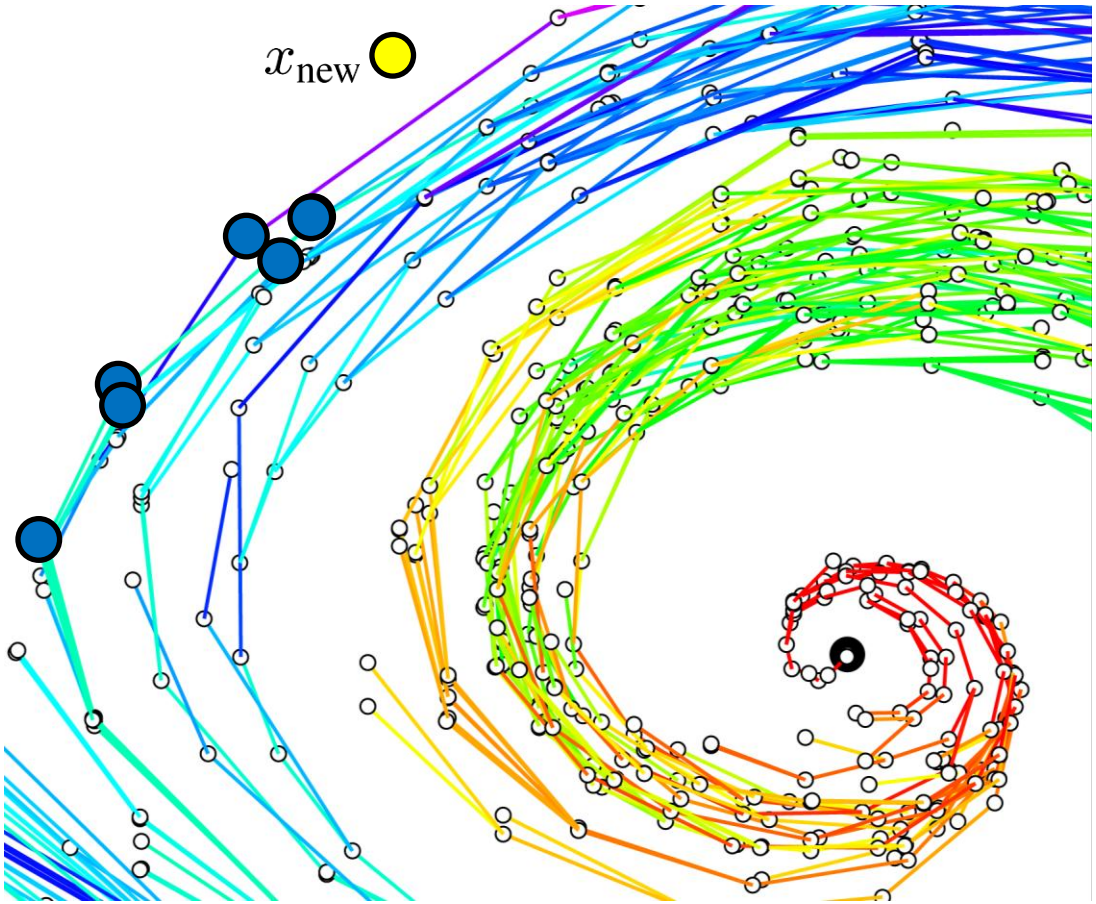
$\text{minCost} \leftarrow \infty$; $x_{\min} \leftarrow \texttt{NULL}$; $\sigma_{\min} \leftarrow \texttt{NULL}$;
for $x_{\text{near}} \in X_{\text{near}}$ **do**
  $\sigma \leftarrow \texttt{LQRSteer}(x_{\text{near}}, x_{\text{new}})$;
  if $\texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma) < \text{minCost}$ **then**
    $\text{minCost} \leftarrow \texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma)$;
    $x_{\min} \leftarrow x_{\text{near}}$; $\sigma_{\min} \leftarrow \sigma$;
return $(x_{\min}, \sigma_{\min})$;

Algorithm 6: $\texttt{Rewire}((V,E), X_{\text{near}}, x_{\text{new}})$

for $x_{\text{near}} \in X_{\text{near}}$ **do**
  $\sigma \leftarrow \texttt{LQRSteer}(x_{\text{new}}, x_{\text{near}})$;
  if $\texttt{Cost}(x_{\text{new}}) + \texttt{Cost}(\sigma) < \texttt{Cost}(x_{\text{near}})$ **then**
    if $\texttt{CollisionFree}(\sigma)$ **then**
      $x_{\text{parent}} \leftarrow \texttt{Parent}(x_{\text{near}})$;
      $E \leftarrow E \setminus \{x_{\text{parent}}, x_{\text{near}}\}$;
      $E \leftarrow E \cup \{x_{\text{new}}, x_{\text{near}}\}$;
return (V,E);

CSAIL

LQR-RRT*



Algorithm 4: $\texttt{LQR}-\texttt{RRT}^*((V,E),N)$

```
for i = 1, ..., N do
    x_rand ← Sample;
    x_nearest ← LQRNearest(V, x_rand);
    x_new ← LQRSteer(x_nearest, x_rand);
    X_near ← LQRNear(V, x_new);
    (x_min, σ_min) ← ChooseParent(X_near, x_new);
    if CollisionFree(σ) then
        X ← X ∪ {x_new};
        E ← E ∪ {(x_min, x_new)};
        (V, E) ← Rewire( (V, E), X_near, x_new );
return G = (V, E);
```

Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

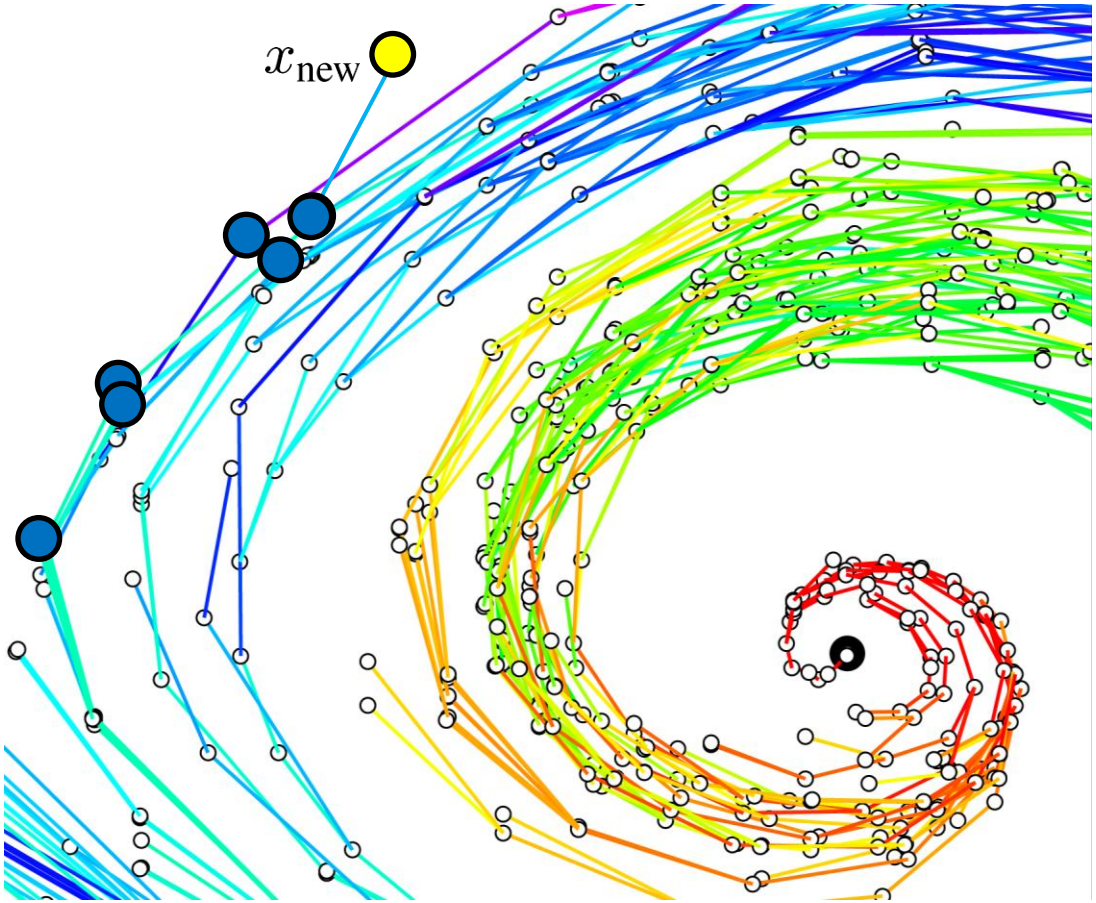
```
minCost ← ∞;  x_min ← NULL;  σ_min ← NULL;
for x_near ∈ X_near do
    σ ← LQRSteer(x_near, x_new);
    if Cost(x_near) + Cost(σ) < minCost then
        minCost ← Cost(x_near) + Cost(σ);
        x_min ← x_near;  σ_min ← σ;
return (x_min, σ_min);
```

Algorithm 6: $\texttt{Rewire}(\,(V,E), X_{\text{near}}, x_{\text{new}}\,)$

```
for x_near ∈ X_near do
    σ ← LQRSteer(x_new, x_near) ;
    if Cost(x_new) + Cost(σ) < Cost(x_near) then
        if CollisionFree(σ) then
            x_parent ← Parent(x_near);
            E ← E \ {x_parent, x_near};
            E ← E ∪ {x_new, x_near};
return (V, E);
```


CSAIL

LQR-RRT*



Algorithm 4: $\texttt{LQR}-\texttt{RRT}^*((V,E),N)$

```
for i = 1, ..., N do
    x_rand ← Sample;
    x_nearest ← LQRNearest(V, x_rand);
    x_new ← LQRSteer(x_nearest, x_rand);
    X_near ← LQRNear(V, x_new);
    (x_min, σ_min) ← ChooseParent(X_near, x_new);
    if CollisionFree(σ) then
        X ← X ∪ {x_new};
        E ← E ∪ {(x_min, x_new)};
        (V, E) ← Rewire( (V, E), X_near, x_new );
return G = (V, E);
```

Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

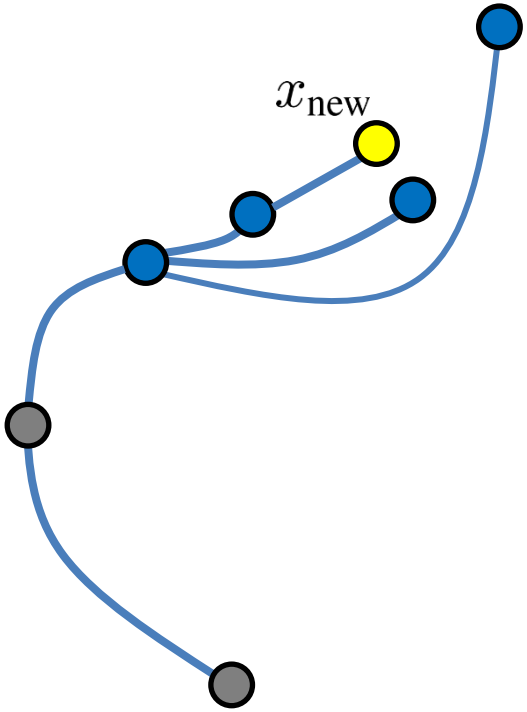
```
minCost ← ∞; x_min ← NULL; σ_min ← NULL;
for x_near ∈ X_near do
    σ ← LQRSteer(x_near, x_new);
    if Cost(x_near) + Cost(σ) < minCost then
        minCost ← Cost(x_near) + Cost(σ);
        x_min ← x_near; σ_min ← σ;
return (x_min, σ_min);
```

Algorithm 6: $\texttt{Rewire}(\,(V,E), X_{\text{near}}, x_{\text{new}}\,)$

```
for x_near ∈ X_near do
    σ ← LQRSteer(x_new, x_near) ;
    if Cost(x_new) + Cost(σ) < Cost(x_near) then
        if CollisionFree(σ) then
            x_parent ← Parent(x_near);
            E ← E \ {x_parent, x_near};
            E ← E ∪ {x_new, x_near};
return (V, E);
```


CSAIL

LQR-RRT*



Algorithm 4: $\texttt{LQR} - \texttt{RRT}^*((V, E), N)$

for $i = 1, \ldots, N$ **do**
  $x_{\text{rand}} \leftarrow \texttt{Sample}$;
  $x_{\text{nearest}} \leftarrow \texttt{LQRNearest}(V, x_{\text{rand}})$;
  $x_{\text{new}} \leftarrow \texttt{LQRSteer}(x_{\text{nearest}}, x_{\text{rand}})$;
  $X_{\text{near}} \leftarrow \texttt{LQRNear}(V, x_{\text{new}})$;
  $(x_{\min}, \sigma_{\min}) \leftarrow \texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$;
  if $\texttt{CollisionFree}(\sigma)$ **then**
    $X \leftarrow X \cup \{x_{\text{new}}\}$;
    $E \leftarrow E \cup \{(x_{\min}, x_{\text{new}})\}$;
    $(V, E) \leftarrow \texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$;
return $G = (V, E)$;

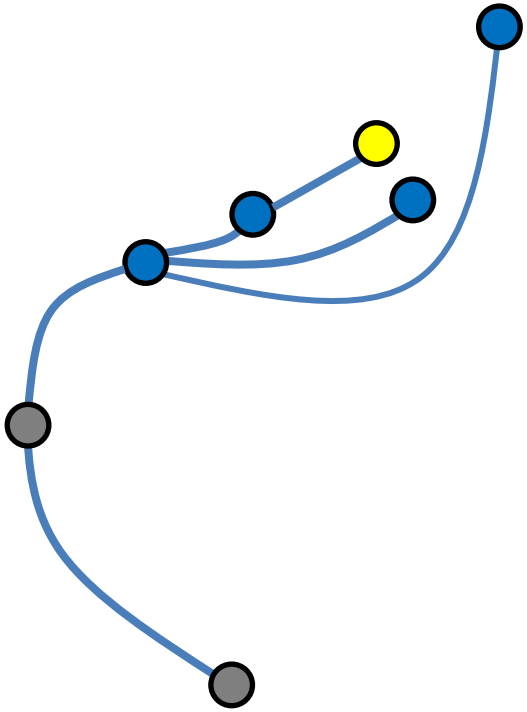
Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

$\text{minCost} \leftarrow \infty$; $x_{\min} \leftarrow \texttt{NULL}$; $\sigma_{\min} \leftarrow \texttt{NULL}$;
for $x_{\text{near}} \in X_{\text{near}}$ **do**
  $\sigma \leftarrow \texttt{LQRSteer}(x_{\text{near}}, x_{\text{new}})$;
  if $\texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma) < \text{minCost}$ **then**
    $\text{minCost} \leftarrow \texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma)$;
    $x_{\min} \leftarrow x_{\text{near}}$; $\sigma_{\min} \leftarrow \sigma$;
return $(x_{\min}, \sigma_{\min})$;

Algorithm 6: $\texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$

for $x_{\text{near}} \in X_{\text{near}}$ **do**
  $\sigma \leftarrow \texttt{LQRSteer}(x_{\text{new}}, x_{\text{near}})$;
  if $\texttt{Cost}(x_{\text{new}}) + \texttt{Cost}(\sigma) < \texttt{Cost}(x_{\text{near}})$ **then**
    if $\texttt{CollisionFree}(\sigma)$ **then**
      $x_{\text{parent}} \leftarrow \texttt{Parent}(x_{\text{near}})$;
      $E \leftarrow E \setminus \{x_{\text{parent}}, x_{\text{near}}\}$;
      $E \leftarrow E \cup \{x_{\text{new}}, x_{\text{near}}\}$;
return (V, E);

LQR-RRT*



Algorithm 4: $\texttt{LQR} - \texttt{RRT}^*((V, E), N)$

1 **for** $i = 1, \ldots, N$ **do**
2 $\quad x_{\text{rand}} \leftarrow \texttt{Sample}$;
3 $\quad x_{\text{nearest}} \leftarrow \texttt{LQRNearest}(V, x_{\text{rand}})$;
4 $\quad x_{\text{new}} \leftarrow \texttt{LQRSteer}(x_{\text{nearest}}, x_{\text{rand}})$;
5 $\quad X_{\text{near}} \leftarrow \texttt{LQRNear}(V, x_{\text{new}})$;
6 $\quad (x_{\min}, \sigma_{\min}) \leftarrow \texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$;
7 $\quad$ **if** $\texttt{CollisionFree}(\sigma)$ **then**
8 $\quad\quad X \leftarrow X \cup \{x_{\text{new}}\}$;
9 $\quad\quad E \leftarrow E \cup \{(x_{\min}, x_{\text{new}})\}$;
10 $\quad\quad (V, E) \leftarrow \texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$;
11 **return** $G = (V, E)$;

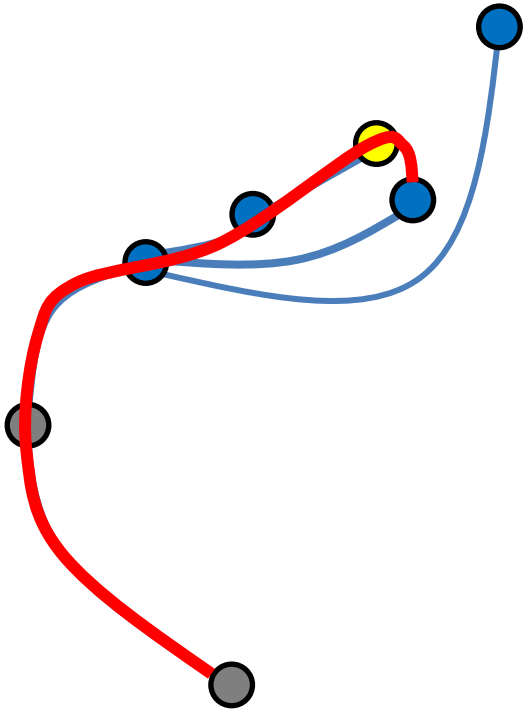
Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

1 $\text{minCost} \leftarrow \infty$; $x_{\min} \leftarrow \texttt{NULL}$; $\sigma_{\min} \leftarrow \texttt{NULL}$;
2 **for** $x_{\text{near}} \in X_{\text{near}}$ **do**
3 $\quad \sigma \leftarrow \texttt{LQRSteer}(x_{\text{near}}, x_{\text{new}})$;
4 $\quad$ **if** $\texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma) < \text{minCost}$ **then**
5 $\quad\quad \text{minCost} \leftarrow \texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma)$;
6 $\quad\quad x_{\min} \leftarrow x_{\text{near}}$; $\sigma_{\min} \leftarrow \sigma$;
7 **return** $(x_{\min}, \sigma_{\min})$;

Algorithm 6: $\texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$

1 **for** $x_{\text{near}} \in X_{\text{near}}$ **do**
2 $\quad \sigma \leftarrow \texttt{LQRSteer}(x_{\text{new}}, x_{\text{near}})$;
3 $\quad$ **if** $\texttt{Cost}(x_{\text{new}}) + \texttt{Cost}(\sigma) < \texttt{Cost}(x_{\text{near}})$ **then**
4 $\quad\quad$ **if** $\texttt{CollisionFree}(\sigma)$ **then**
5 $\quad\quad\quad x_{\text{parent}} \leftarrow \texttt{Parent}(x_{\text{near}})$;
6 $\quad\quad\quad E \leftarrow E \setminus \{x_{\text{parent}}, x_{\text{near}}\}$;
7 $\quad\quad\quad E \leftarrow E \cup \{x_{\text{new}}, x_{\text{near}}\}$;
8 **return** (V, E);

LQR-RRT*



Algorithm 4: $\texttt{LQR} - \texttt{RRT}^*((V, E), N)$

1 **for** $i = 1, \ldots, N$ **do**
2 $\quad x_{\text{rand}} \leftarrow \texttt{Sample}$;
3 $\quad x_{\text{nearest}} \leftarrow \texttt{LQRNearest}(V, x_{\text{rand}})$;
4 $\quad x_{\text{new}} \leftarrow \texttt{LQRSteer}(x_{\text{nearest}}, x_{\text{rand}})$;
5 $\quad X_{\text{near}} \leftarrow \texttt{LQRNear}(V, x_{\text{new}})$;
6 $\quad (x_{\min}, \sigma_{\min}) \leftarrow \texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$;
7 $\quad$ **if** $\texttt{CollisionFree}(\sigma)$ **then**
8 $\quad\quad X \leftarrow X \cup \{x_{\text{new}}\}$;
9 $\quad\quad E \leftarrow E \cup \{(x_{\min}, x_{\text{new}})\}$;
10 $\quad\quad (V, E) \leftarrow \texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$;
11 **return** $G = (V, E)$;

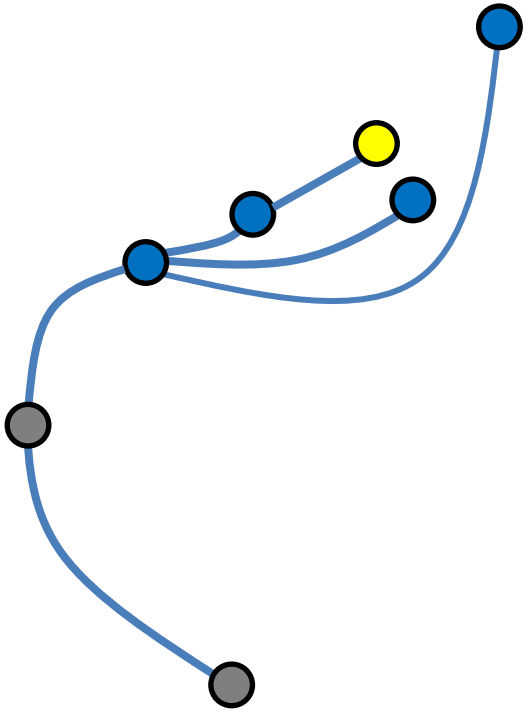
Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

1 $\text{minCost} \leftarrow \infty$; $x_{\min} \leftarrow \texttt{NULL}$; $\sigma_{\min} \leftarrow \texttt{NULL}$;
2 **for** $x_{\text{near}} \in X_{\text{near}}$ **do**
3 $\quad \sigma \leftarrow \texttt{LQRSteer}(x_{\text{near}}, x_{\text{new}})$;
4 $\quad$ **if** $\texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma) < \text{minCost}$ **then**
5 $\quad\quad \text{minCost} \leftarrow \texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma)$;
6 $\quad\quad x_{\min} \leftarrow x_{\text{near}}$; $\sigma_{\min} \leftarrow \sigma$;
7 **return** $(x_{\min}, \sigma_{\min})$;

Algorithm 6: $\texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$

1 **for** $x_{\text{near}} \in X_{\text{near}}$ **do**
2 $\quad \sigma \leftarrow \texttt{LQRSteer}(x_{\text{new}}, x_{\text{near}})$;
3 $\quad$ **if** $\texttt{Cost}(x_{\text{new}}) + \texttt{Cost}(\sigma) < \texttt{Cost}(x_{\text{near}})$ **then**
4 $\quad\quad$ **if** $\texttt{CollisionFree}(\sigma)$ **then**
5 $\quad\quad\quad x_{\text{parent}} \leftarrow \texttt{Parent}(x_{\text{near}})$;
6 $\quad\quad\quad E \leftarrow E \setminus \{x_{\text{parent}}, x_{\text{near}}\}$;
7 $\quad\quad\quad E \leftarrow E \cup \{x_{\text{new}}, x_{\text{near}}\}$;
8 **return** (V, E);

LQR-RRT*



Algorithm 4: $\texttt{LQR} - \texttt{RRT}^*((V, E), N)$

1 **for** $i = 1, \ldots, N$ **do**
2 $\quad x_{\text{rand}} \leftarrow \texttt{Sample}$;
3 $\quad x_{\text{nearest}} \leftarrow \texttt{LQRNearest}(V, x_{\text{rand}})$;
4 $\quad x_{\text{new}} \leftarrow \texttt{LQRSteer}(x_{\text{nearest}}, x_{\text{rand}})$;
5 $\quad X_{\text{near}} \leftarrow \texttt{LQRNear}(V, x_{\text{new}})$;
6 $\quad (x_{\min}, \sigma_{\min}) \leftarrow \texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$;
7 $\quad$ **if** $\texttt{CollisionFree}(\sigma)$ **then**
8 $\quad\quad X \leftarrow X \cup \{x_{\text{new}}\}$;
9 $\quad\quad E \leftarrow E \cup \{(x_{\min}, x_{\text{new}})\}$;
10 $\quad\quad (V, E) \leftarrow \texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$;
11 **return** $G = (V, E)$;

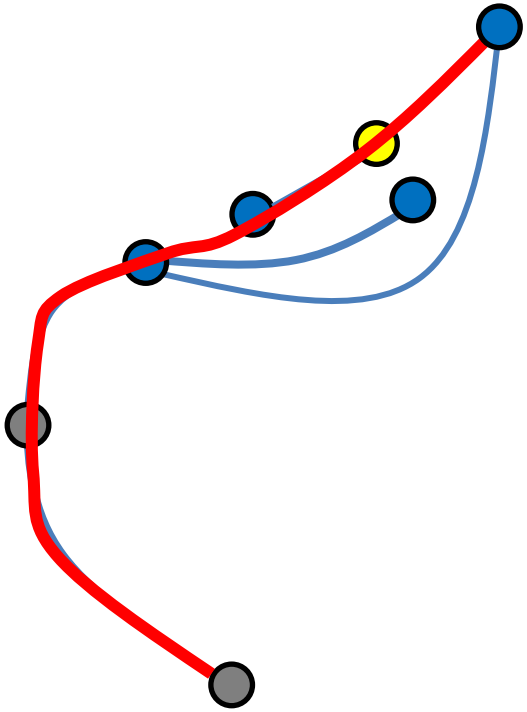
Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

1 $\text{minCost} \leftarrow \infty$; $x_{\min} \leftarrow \texttt{NULL}$; $\sigma_{\min} \leftarrow \texttt{NULL}$;
2 **for** $x_{\text{near}} \in X_{\text{near}}$ **do**
3 $\quad \sigma \leftarrow \texttt{LQRSteer}(x_{\text{near}}, x_{\text{new}})$;
4 $\quad$ **if** $\texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma) < \text{minCost}$ **then**
5 $\quad\quad \text{minCost} \leftarrow \texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma)$;
6 $\quad\quad x_{\min} \leftarrow x_{\text{near}}$; $\sigma_{\min} \leftarrow \sigma$;
7 **return** $(x_{\min}, \sigma_{\min})$;

Algorithm 6: $\texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$

1 **for** $x_{\text{near}} \in X_{\text{near}}$ **do**
2 $\quad \sigma \leftarrow \texttt{LQRSteer}(x_{\text{new}}, x_{\text{near}})$;
3 $\quad$ **if** $\texttt{Cost}(x_{\text{new}}) + \texttt{Cost}(\sigma) < \texttt{Cost}(x_{\text{near}})$ **then**
4 $\quad\quad$ **if** $\texttt{CollisionFree}(\sigma)$ **then**
5 $\quad\quad\quad x_{\text{parent}} \leftarrow \texttt{Parent}(x_{\text{near}})$;
6 $\quad\quad\quad E \leftarrow E \setminus \{x_{\text{parent}}, x_{\text{near}}\}$;
7 $\quad\quad\quad E \leftarrow E \cup \{x_{\text{new}}, x_{\text{near}}\}$;
8 **return** (V, E);

LQR-RRT*



Algorithm 4: $\texttt{LQR} - \texttt{RRT}^*((V, E), N)$

1 **for** $i = 1, \ldots, N$ **do**
2 $\quad x_{\text{rand}} \leftarrow \texttt{Sample}$;
3 $\quad x_{\text{nearest}} \leftarrow \texttt{LQRNearest}(V, x_{\text{rand}})$;
4 $\quad x_{\text{new}} \leftarrow \texttt{LQRSteer}(x_{\text{nearest}}, x_{\text{rand}})$;
5 $\quad X_{\text{near}} \leftarrow \texttt{LQRNear}(V, x_{\text{new}})$;
6 $\quad (x_{\min}, \sigma_{\min}) \leftarrow \texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$;
7 $\quad$ **if** $\texttt{CollisionFree}(\sigma)$ **then**
8 $\quad\quad X \leftarrow X \cup \{x_{\text{new}}\}$;
9 $\quad\quad E \leftarrow E \cup \{(x_{\min}, x_{\text{new}})\}$;
10 $\quad\quad (V, E) \leftarrow \texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$;
11 **return** $G = (V, E)$;

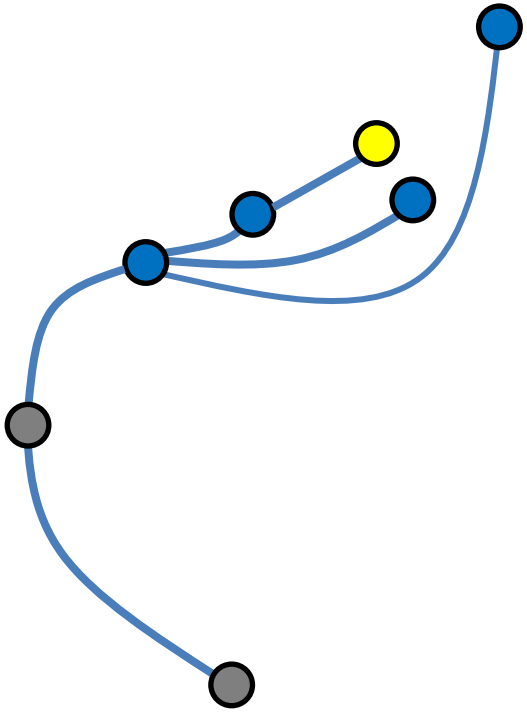
Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

1 $\text{minCost} \leftarrow \infty$; $x_{\min} \leftarrow \texttt{NULL}$; $\sigma_{\min} \leftarrow \texttt{NULL}$;
2 **for** $x_{\text{near}} \in X_{\text{near}}$ **do**
3 $\quad \sigma \leftarrow \texttt{LQRSteer}(x_{\text{near}}, x_{\text{new}})$;
4 $\quad$ **if** $\texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma) < \text{minCost}$ **then**
5 $\quad\quad \text{minCost} \leftarrow \texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma)$;
6 $\quad\quad x_{\min} \leftarrow x_{\text{near}}$; $\sigma_{\min} \leftarrow \sigma$;
7 **return** $(x_{\min}, \sigma_{\min})$;

Algorithm 6: $\texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$

1 **for** $x_{\text{near}} \in X_{\text{near}}$ **do**
2 $\quad \sigma \leftarrow \texttt{LQRSteer}(x_{\text{new}}, x_{\text{near}})$;
3 $\quad$ **if** $\texttt{Cost}(x_{\text{new}}) + \texttt{Cost}(\sigma) < \texttt{Cost}(x_{\text{near}})$ **then**
4 $\quad\quad$ **if** $\texttt{CollisionFree}(\sigma)$ **then**
5 $\quad\quad\quad x_{\text{parent}} \leftarrow \texttt{Parent}(x_{\text{near}})$;
6 $\quad\quad\quad E \leftarrow E \setminus \{x_{\text{parent}}, x_{\text{near}}\}$;
7 $\quad\quad\quad E \leftarrow E \cup \{x_{\text{new}}, x_{\text{near}}\}$;
8 **return** (V, E);

LQR-RRT*



Algorithm 4: $\texttt{LQR} - \texttt{RRT}^*((V, E), N)$

1 **for** $i = 1, \ldots, N$ **do**
2 $\quad x_{\text{rand}} \leftarrow \texttt{Sample}$;
3 $\quad x_{\text{nearest}} \leftarrow \texttt{LQRNearest}(V, x_{\text{rand}})$;
4 $\quad x_{\text{new}} \leftarrow \texttt{LQRSteer}(x_{\text{nearest}}, x_{\text{rand}})$;
5 $\quad X_{\text{near}} \leftarrow \texttt{LQRNear}(V, x_{\text{new}})$;
6 $\quad (x_{\min}, \sigma_{\min}) \leftarrow \texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$;
7 $\quad$ **if** $\texttt{CollisionFree}(\sigma)$ **then**
8 $\quad\quad X \leftarrow X \cup \{x_{\text{new}}\}$;
9 $\quad\quad E \leftarrow E \cup \{(x_{\min}, x_{\text{new}})\}$;
10 $\quad\quad (V, E) \leftarrow \texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$;
11 **return** $G = (V, E)$;

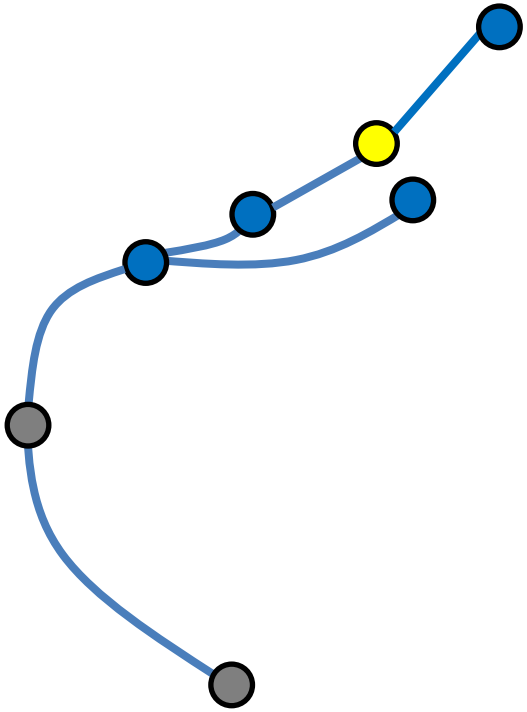
Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

1 $\text{minCost} \leftarrow \infty$; $x_{\min} \leftarrow \texttt{NULL}$; $\sigma_{\min} \leftarrow \texttt{NULL}$;
2 **for** $x_{\text{near}} \in X_{\text{near}}$ **do**
3 $\quad \sigma \leftarrow \texttt{LQRSteer}(x_{\text{near}}, x_{\text{new}})$;
4 $\quad$ **if** $\texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma) < \text{minCost}$ **then**
5 $\quad\quad \text{minCost} \leftarrow \texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma)$;
6 $\quad\quad x_{\min} \leftarrow x_{\text{near}}$; $\sigma_{\min} \leftarrow \sigma$;
7 **return** $(x_{\min}, \sigma_{\min})$;

Algorithm 6: $\texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$

1 **for** $x_{\text{near}} \in X_{\text{near}}$ **do**
2 $\quad \sigma \leftarrow \texttt{LQRSteer}(x_{\text{new}}, x_{\text{near}})$;
3 $\quad$ **if** $\texttt{Cost}(x_{\text{new}}) + \texttt{Cost}(\sigma) < \texttt{Cost}(x_{\text{near}})$ **then**
4 $\quad\quad$ **if** $\texttt{CollisionFree}(\sigma)$ **then**
5 $\quad\quad\quad x_{\text{parent}} \leftarrow \texttt{Parent}(x_{\text{near}})$;
6 $\quad\quad\quad E \leftarrow E \setminus \{x_{\text{parent}}, x_{\text{near}}\}$;
7 $\quad\quad\quad E \leftarrow E \cup \{x_{\text{new}}, x_{\text{near}}\}$;
8 **return** (V, E);

LQR-RRT*



Algorithm 4: $\texttt{LQR} - \texttt{RRT}^*((V, E), N)$

1 **for** $i = 1, \ldots, N$ **do**
2 $\quad x_{\text{rand}} \leftarrow \texttt{Sample}$;
3 $\quad x_{\text{nearest}} \leftarrow \texttt{LQRNearest}(V, x_{\text{rand}})$;
4 $\quad x_{\text{new}} \leftarrow \texttt{LQRSteer}(x_{\text{nearest}}, x_{\text{rand}})$;
5 $\quad X_{\text{near}} \leftarrow \texttt{LQRNear}(V, x_{\text{new}})$;
6 $\quad (x_{\min}, \sigma_{\min}) \leftarrow \texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$;
7 $\quad$ **if** $\texttt{CollisionFree}(\sigma)$ **then**
8 $\quad\quad X \leftarrow X \cup \{x_{\text{new}}\}$;
9 $\quad\quad E \leftarrow E \cup \{(x_{\min}, x_{\text{new}})\}$;
10 $\quad\quad (V, E) \leftarrow \texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$;
11 **return** $G = (V, E)$;

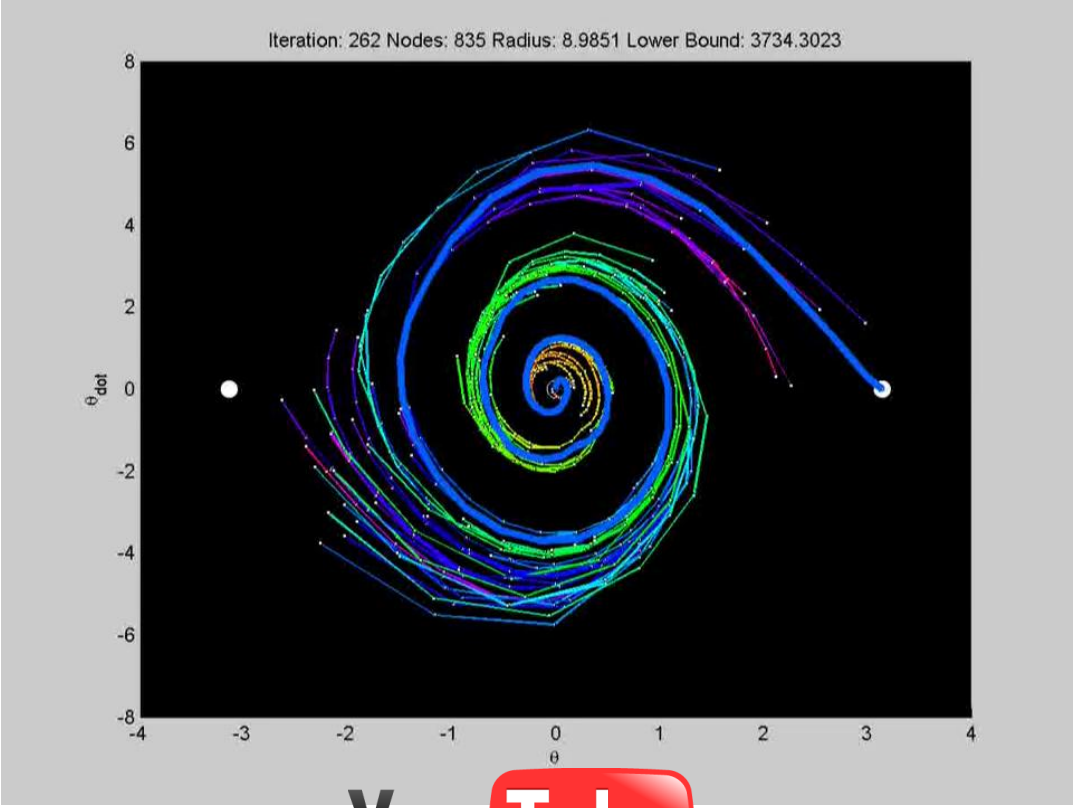
Algorithm 5: $\texttt{ChooseParent}(X_{\text{near}}, x_{\text{new}})$

1 $\text{minCost} \leftarrow \infty$; $x_{\min} \leftarrow \texttt{NULL}$; $\sigma_{\min} \leftarrow \texttt{NULL}$;
2 **for** $x_{\text{near}} \in X_{\text{near}}$ **do**
3 $\quad \sigma \leftarrow \texttt{LQRSteer}(x_{\text{near}}, x_{\text{new}})$;
4 $\quad$ **if** $\texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma) < \text{minCost}$ **then**
5 $\quad\quad \text{minCost} \leftarrow \texttt{Cost}(x_{\text{near}}) + \texttt{Cost}(\sigma)$;
6 $\quad\quad x_{\min} \leftarrow x_{\text{near}}$; $\sigma_{\min} \leftarrow \sigma$;
7 **return** $(x_{\min}, \sigma_{\min})$;

Algorithm 6: $\texttt{Rewire}((V, E), X_{\text{near}}, x_{\text{new}})$

1 **for** $x_{\text{near}} \in X_{\text{near}}$ **do**
2 $\quad \sigma \leftarrow \texttt{LQRSteer}(x_{\text{new}}, x_{\text{near}})$;
3 $\quad$ **if** $\texttt{Cost}(x_{\text{new}}) + \texttt{Cost}(\sigma) < \texttt{Cost}(x_{\text{near}})$ **then**
4 $\quad\quad$ **if** $\texttt{CollisionFree}(\sigma)$ **then**
5 $\quad\quad\quad x_{\text{parent}} \leftarrow \texttt{Parent}(x_{\text{near}})$;
6 $\quad\quad\quad E \leftarrow E \setminus \{x_{\text{parent}}, x_{\text{near}}\}$;
7 $\quad\quad\quad E \leftarrow E \cup \{x_{\text{new}}, x_{\text{near}}\}$;
8 **return** (V, E);

LQR-RRT*

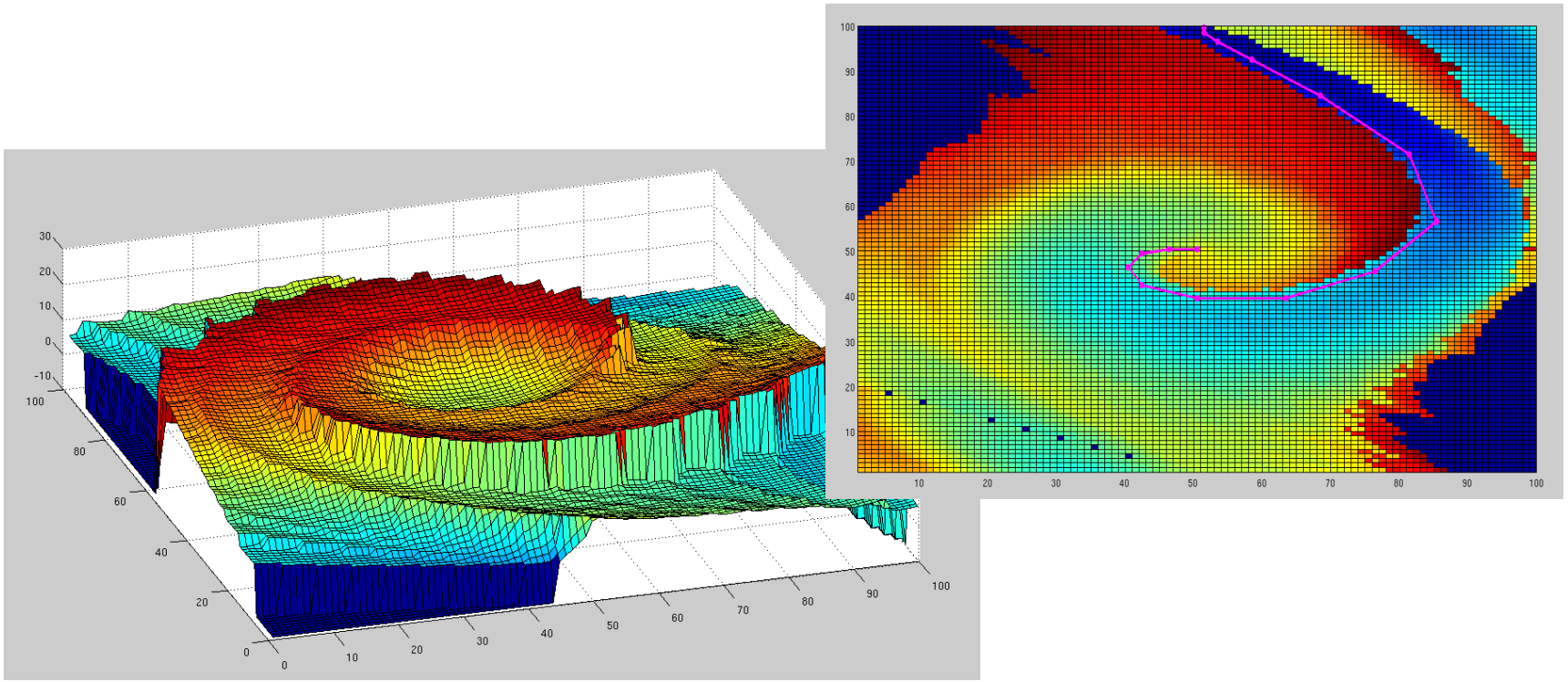


YouTube™

Results: Torque-limited Pendulum (Dynamics with constraints)

Two-dimensional state space, $(\theta, \dot{\theta})$

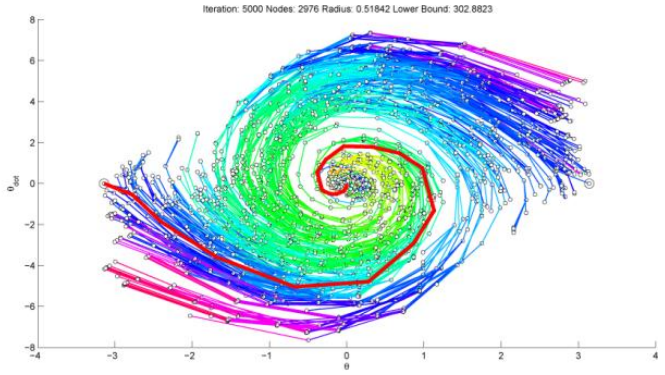
$$\ddot{\theta} = u - b\dot{\theta} - g\cos(\theta),\ g = 9.81,\ b = 0.1,\ u \in [-3, 3]$$



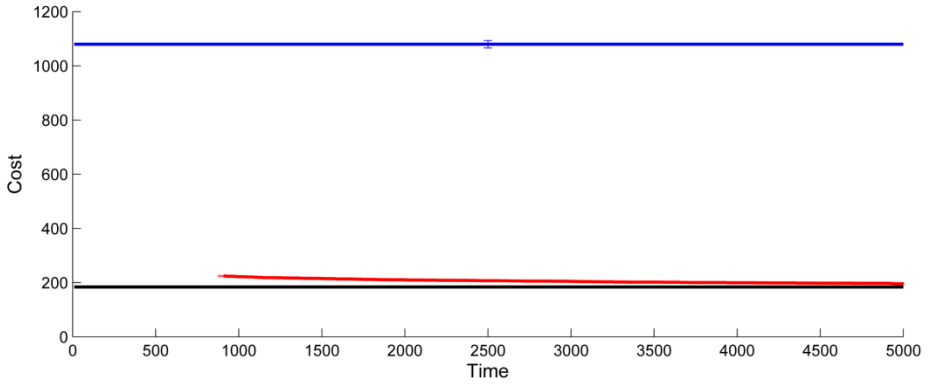
Results: Torque-limited Pendulum (Dynamics with constraints)

Two-dimensional state space, $(\theta, \dot{\theta})$

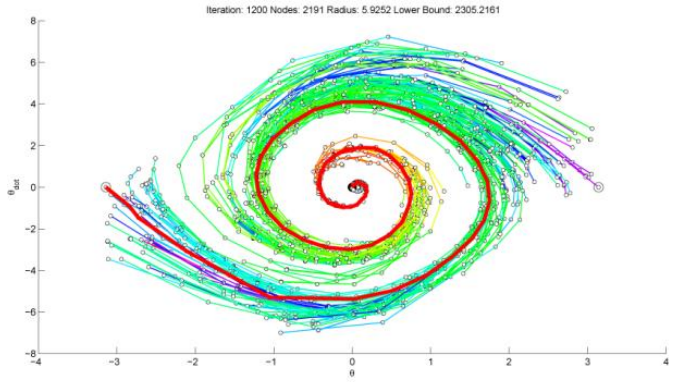
$$\ddot{\theta} = u - b\dot{\theta} - g\cos(\theta),\ g = 9.81,\ b = 0.1,\ u \in [-3, 3]$$



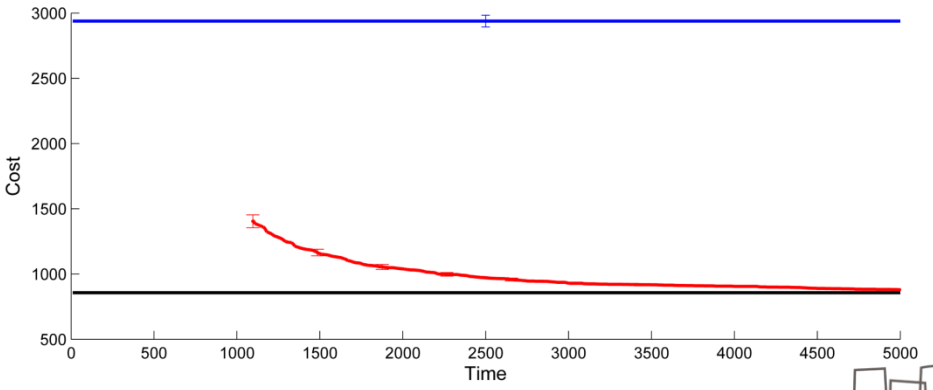
(a) $R = 1$



(a) Torque-limited Pendulum $R = 1$



(b) $R = 50$



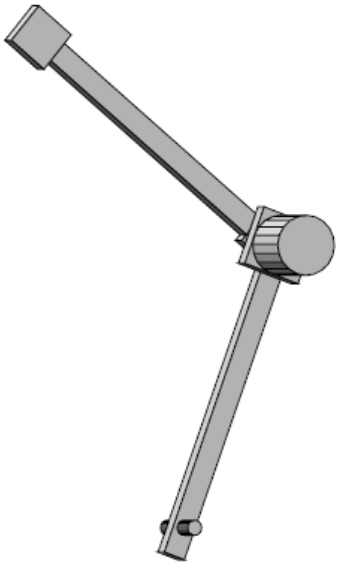
(b) Torque-limited Pendulum $R = 50$

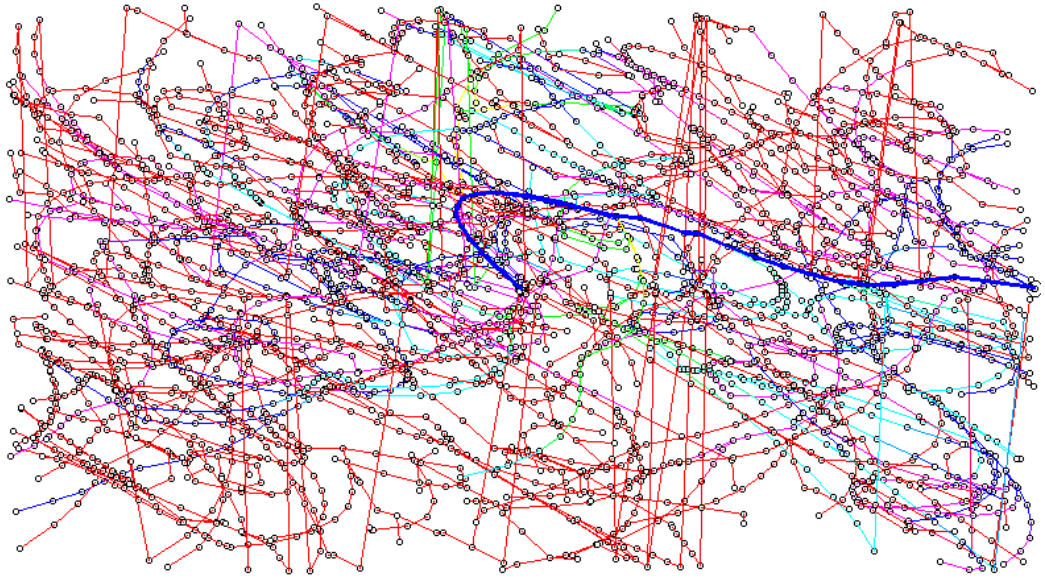
CSAIL

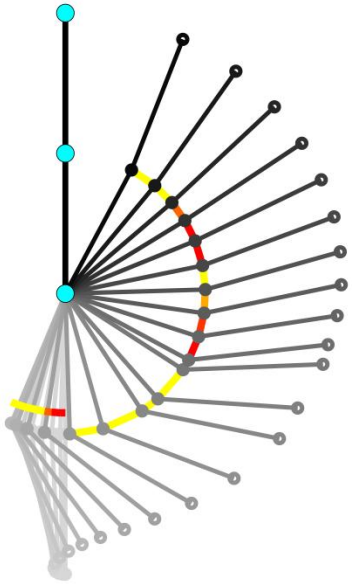
Results: Acrobot (Underactuated Dynamics)

Four-dimensional state space $(q_1, q_2, \dot{q}_1, \dot{q}_2)$

R. Murray and J. Hauser, "A case study in approximate linearization: The acrobot example," EECS Department, University of California, Berkeley, Tech. Rep. UCB/ERL M91/46, 1991.



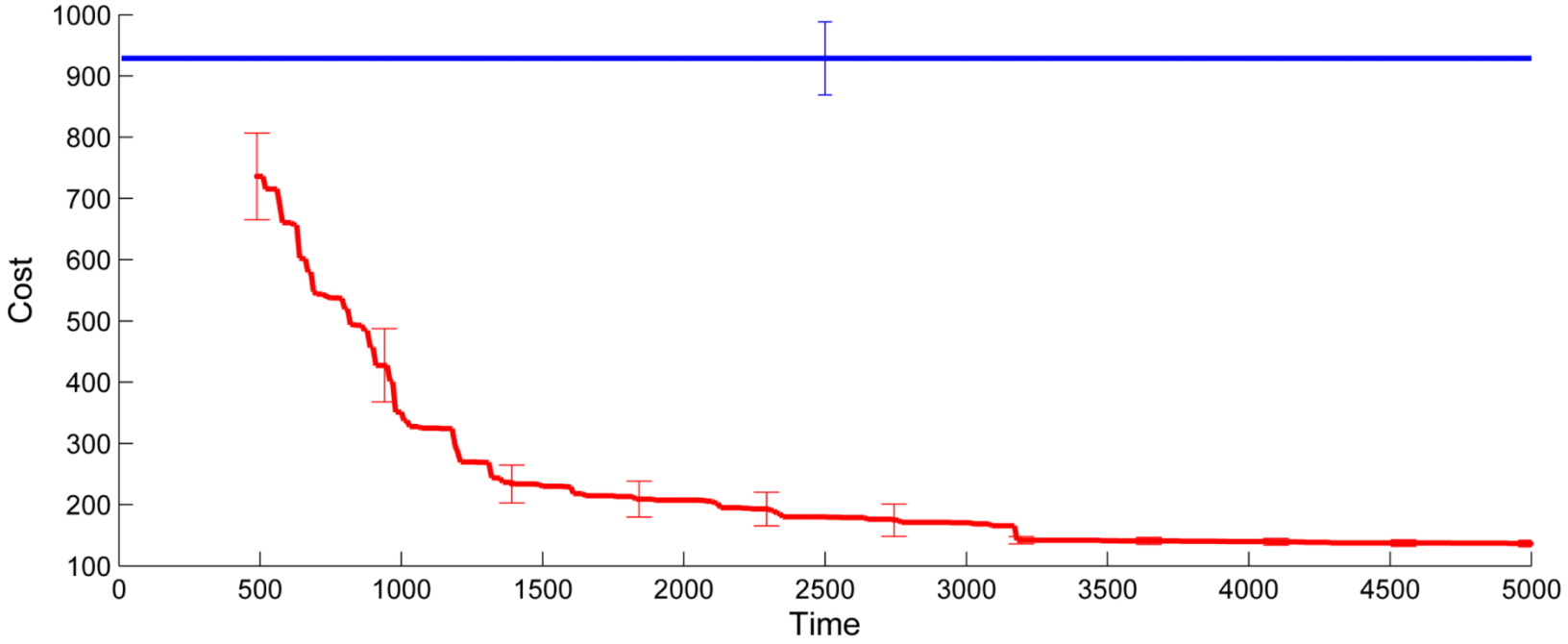




Results: Acrobot (Underactuated Dynamics)

Four-dimensional state space $(q_1, q_2, \dot{q}_1, \dot{q}_2)$

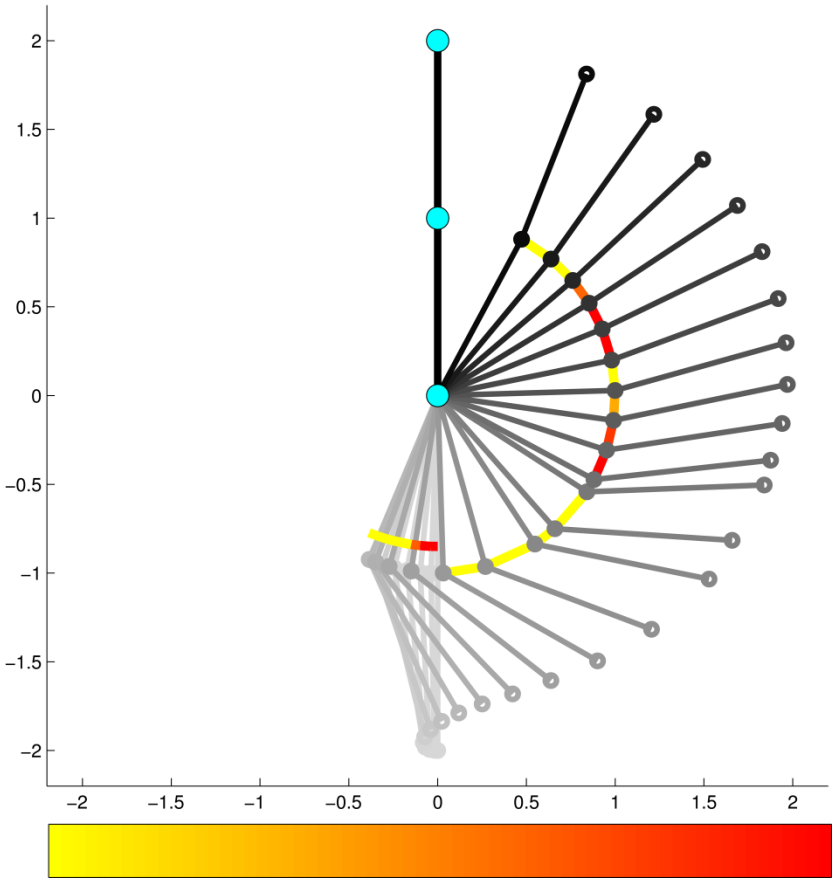
R. Murray and J. Hauser, "A case study in approximate linearization: The acrobot example," EECS Department, University of California, Berkeley, Tech. Rep. UCB/ERL M91/46, 1991.



Results: Acrobot (Underactuated Dynamics)

Four-dimensional state space $(q_1, q_2, \dot{q}_1, \dot{q}_2)$

R. Murray and J. Hauser, "A case study in approximate linearization: The acrobot example," EECS Department, University of California, Berkeley, Tech. Rep. UCB/ERL M91/46, 1991.



Results: Light-Dark Domain (Planning in Belief-Space)

Three-dimensional state space (x_1, x_2, v)

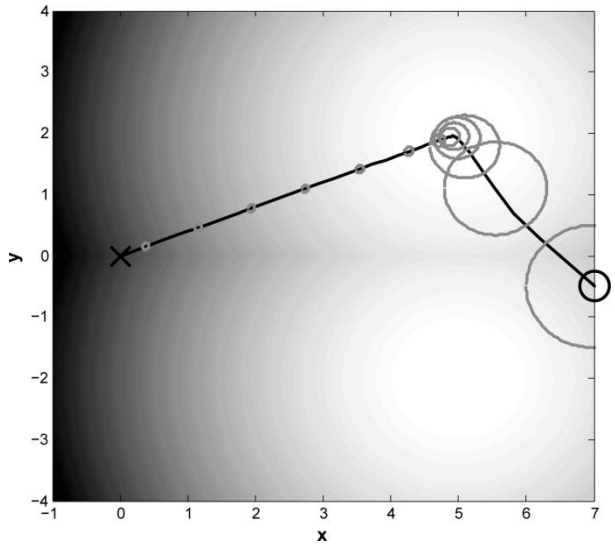
$$w(\mathbf{x}) = \min((\mathbf{x} - \mathbf{b}_1)^2, (\mathbf{x} - \mathbf{b}_2)^2),$$

$$\dot{\mathbf{s}} = (u_1, u_2, \dot{v})^T$$

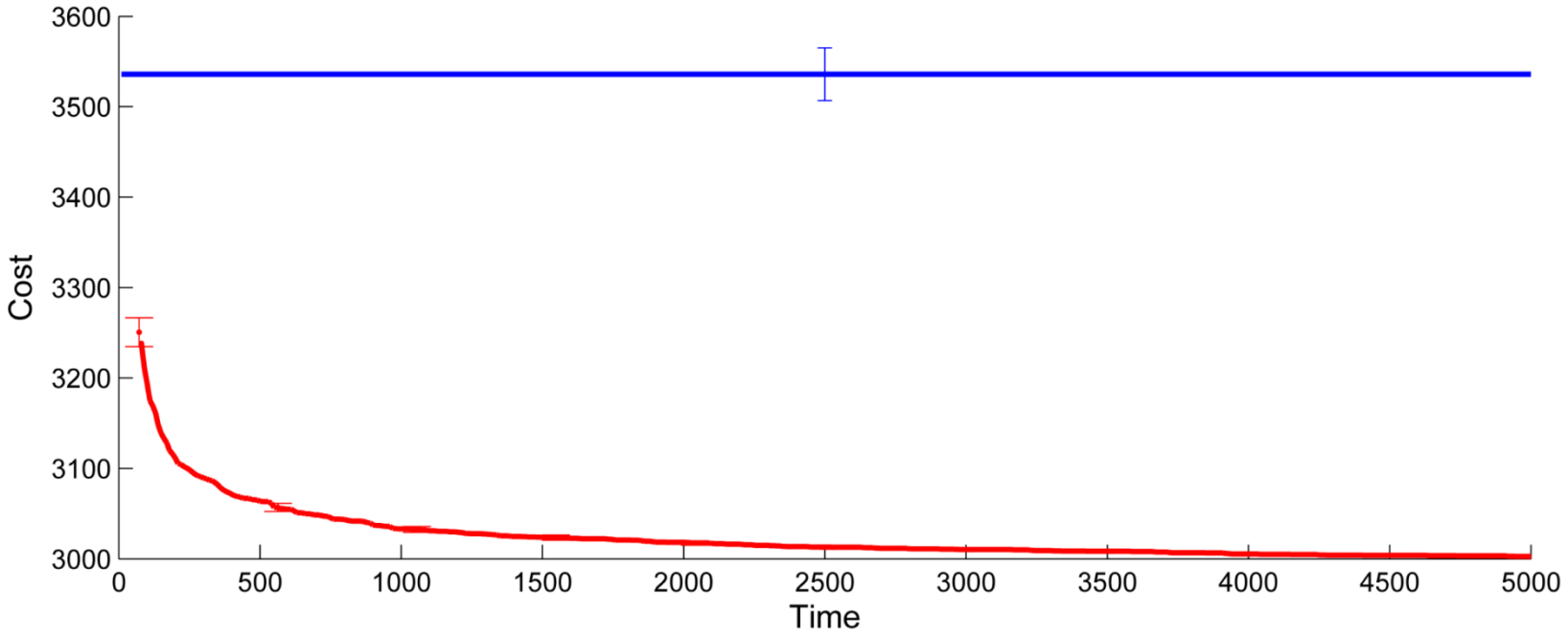
$$\dot{v} = -\frac{v^2}{v + w(\mathbf{x})}$$

Adapted from

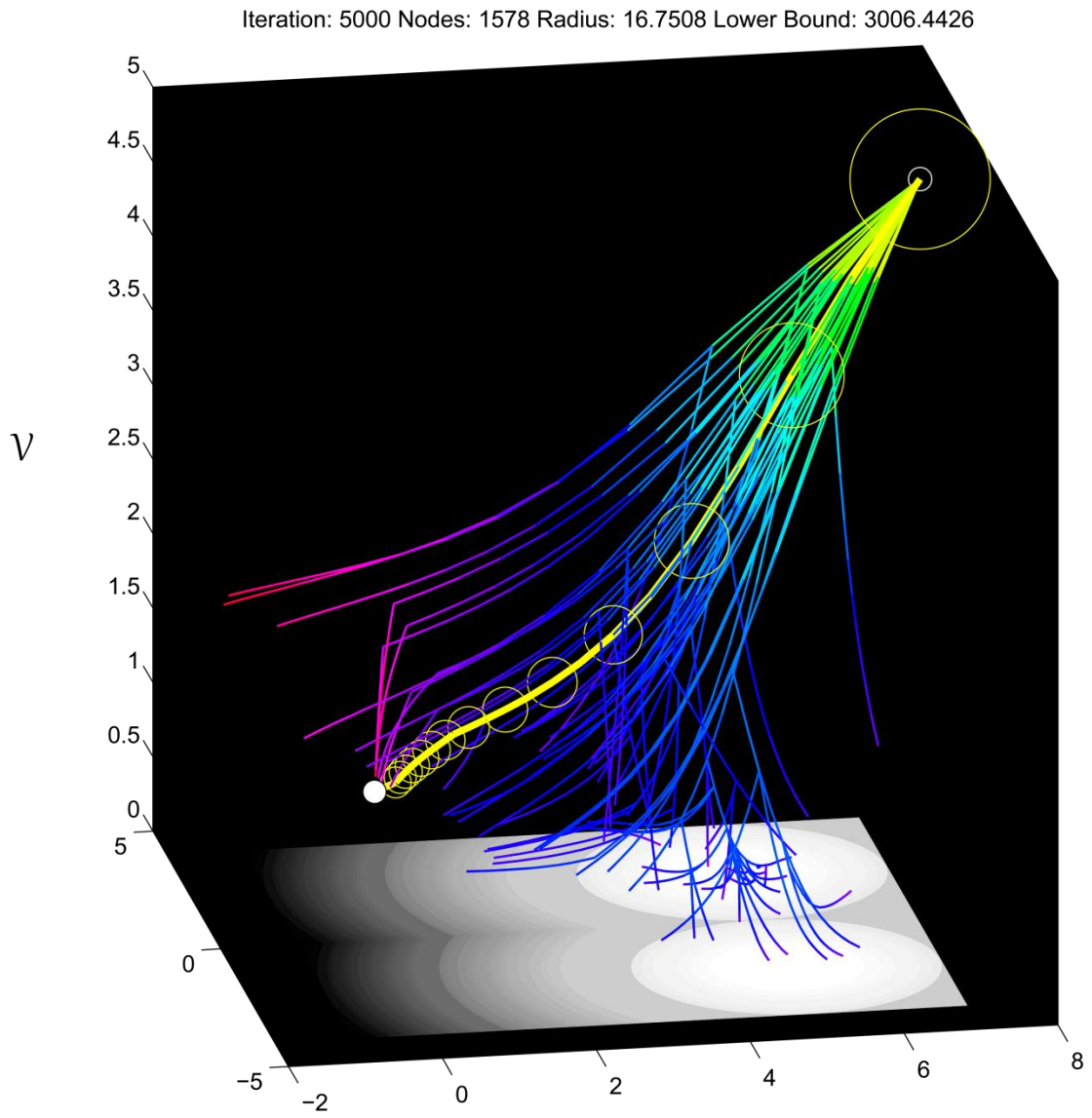
R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez, "Belief space planning assuming maximum likelihood observations," in *Proceedings of Robotics: Science and Systems*, June 2010.



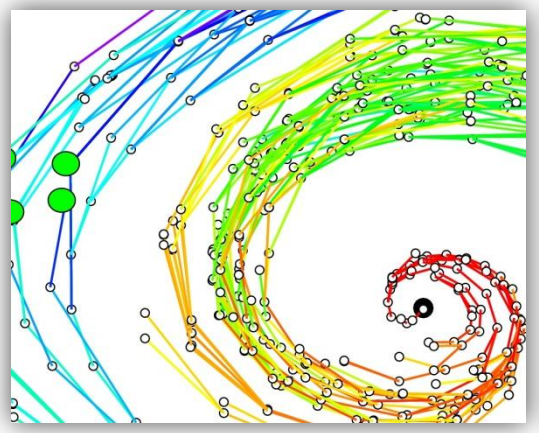
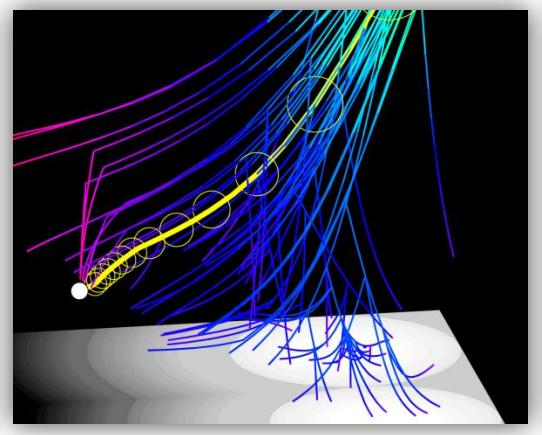
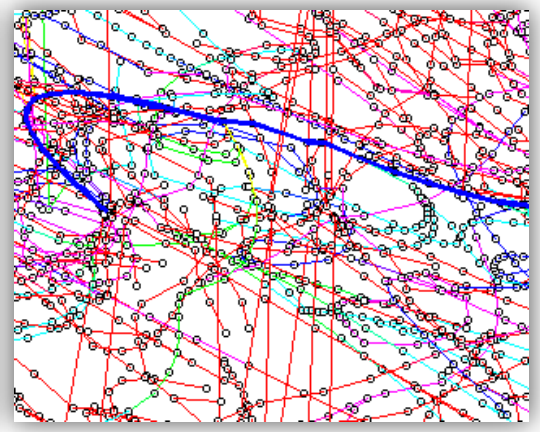
Results: Light-Dark Domain (Planning in Belief-Space)



Results: Light-Dark Domain (Planning in Belief-Space)

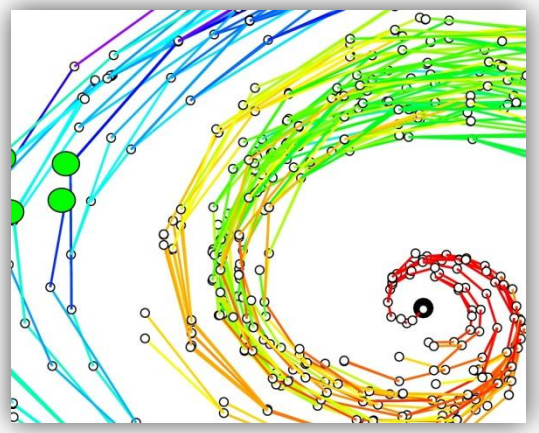
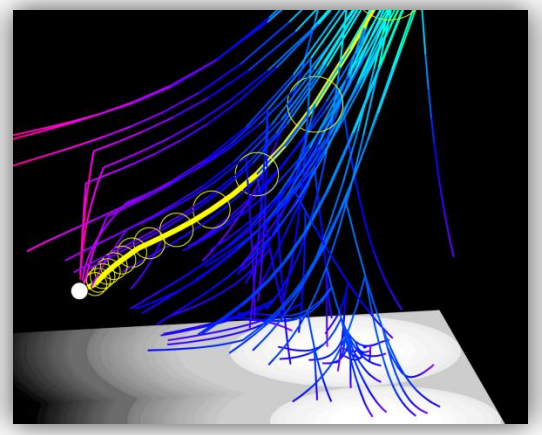
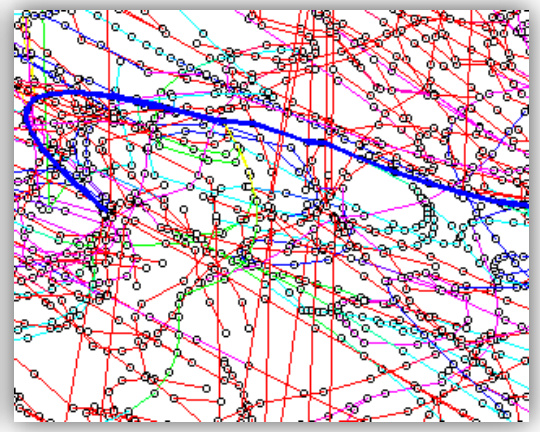


Current Work

- AQR, trajectory optimization by Gustavo Goretkin
- Theoretical Analysis
- Higher Dimensional Domains, Obstacles/Collision Checking

Thanks

For more information:

http://people.csail.mit.edu/aperez/www/

http://lis.csail.mit.edu