The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Feedback Motion Planning Approach for Nonlinear Control using Gain Scheduled RRTs

Guilherme J. Maeda, Surya P. N. Singh, Hugh Durrant-Whyte
Australian Centre for Field Robotics, University of Sydney, Australia
{g.maeda, spns, hugh}@acfr.usyd.edu.au

***Abstract*— A new control strategy based on feedback motion planning is presented for solving nonlinear control problems in constrained environments. The algorithm explores the state-space using a bi-directional rapidly exploring random tree (biRRT) in order to find a feasible trajectory between an initial and goal state. By incrementally scheduling LQR controllers, it attempts to connect states so as to link the two trees. These attempts are evaluated by verifying that the connected state is inside the controllable area of an infinite time horizon controller at the goal. This allows for a rapid delineation of equivalent neighborhoods in the state-space. As a result, random exploration is terminated as soon as a feasible solution is made possible by feedback means, avoiding oversampling and partially introducing optimal actions at the neighborhood of the connection. The algorithm is demonstrated and compared against a biRRT using single-link pendulum and cart-pole swing-up tasks amongst obstacles, the latter showing a nearly order of magnitude more efficient search.**

I. INTRODUCTION

From manipulation to legged hoppers to aerial vehicles, the agile nonlinear kinodynamic robot motion problem has motivated the development of powerful, scalable, sample-based motion planning algorithms. Such methods quickly determine a (feasible) trajectory to reach a goal state. These tools can be extended to tasks beyond trajectory generation, because many tasks, including the design of feedback control laws for nonlinear dynamical systems, can be viewed as a trajectory for the robot to follow (albeit in the state space).

Motion planning methods, such as the Rapidly exploring Randomized Tree (RRT) algorithm [1], typically determine discretized open-loop trajectories that are then presumably tracked using a separate feedback control system. Such a decoupling is a strong assumption that can lead to: (1) trajectories that are difficult (if not impossible) to control, and (2) costly planning around conditions that could have been handled by a controller. This, in turn, argues for an integrated approach where the path is stable and efficiently executable.

Two results from optimal controls provide insight towards feedback motion planning. First, for the Linear Quadratic Regulator (LQR) problem (in the infinite horizon case), controllers can be solved directly (via the Riccati equation) [2] leading to efficient and (under linear conditions) optimal solutions. Second, recent methods (based on convex optimization) allow for the estimation of Lyapunov functions, thus delineating the extent of regions of stability for smooth nonlinear systems [3].

The exploration under differential constraints provides improvements to the RRT by modifying its sampling strategy, for example, by changing its Voronoi size for a non-uniform distribution [4], and by sampling on non-expandable areas [5]. However, little research has addressed the use of feedback during the exploration of the space as a method to handle relations between states. This work introduces, and is based on, the property that a local linear full-state LQR controller has sufficient robustness to generate a connection between tree nodes of a RRT, in the same way it handles set point changes in a regulation problem.

The core idea is that gain scheduling (i.e., a series of linear controllers for local regions of a nonlinear problem) can be integrated to drive trajectory generation by informing which regions of the state-space are "equivalent" since they are within reach of a given controller. In contrast to many feedback motion planning methods, particularly LQR-Trees [6] and navigation function methods [7], there is no intention to cover the greater state-space with control laws as this paper is motivated by single-query nonlinear control cases. The next section introduces and illustrates the Gain Scheduled biRRT (GS-biRRT) using a (torque-limited) inverted pendulum. The latter sections show results for the case of a cart-pole amongst obstacles and indicates improved scalability and efficiency of the method.

II. BACKGROUND AND RELATED WORK

Sampling-based motion planners have been proposed for nonlinear dynamic control problems [5], [8] as an approach for direct search of solutions in the state space. In this context, RRTs [9] were used due to its ability in handling kinodynamic and obstacle constraints.

For holonomic systems, the original RRT is modified such that its EXTEND function (refer to [9] for details of the algorithm) is replaced with a CONNECT function [10]. While the RRT is expanded with a limited (often fixed) step size towards a sample, in this case, the RRT connects to the sample and only stops if an obstacle is reached. However, this is a purely geometrical problem and assumes a trivial inversion between initial and final states, and does not consider dynamic constraints. The GS-biRRT

978-1-4244-6676-4/10/$25.00

uses a similar CONNECT function, but addresses dynamic constraints trough a closed-loop controller.

A thread in feedback motion planning is the idea of representing local stability regions of a controller. This concept may be viewed as a funnel representing a Lyapunov image [11], where a local controller is able to bring any state within the borders of the funnel to the single minima representing a local goal. With an image of a local stability, and using the Lyapunov property as a local navigation function, strategies based on experiments [7], and sampling [12] are applied to create a sequence of funnels representing different states and controller stabilities, whose final output is the single, desired goal. Recently methods for approximating the basin of attraction based on sum-of-squares optimization have been proposed [3], [6]. This allows for the generation of a tree of LQR controllers that are extended to cover the space with the estimated basin of attractions. Such a tool brings the opportunity to explore feedback motion planning strategies in conjunction with sample-based planners.

The method proposed in [6] is used to verify the stability region of a time invariant LQR controller with an infinite time horizon to achieve a goal. As shown in Fig. 1, the funnel (representing a Lyapunov function) illustrates the fact that there is a certain amount of acceptable error during trajectory tracking. If the tracking controller can bring the state x_{sim} inside the the basin of attraction, then an infinite horizon controller can drive x_{sim} to x_{goal} as time goes to infinity.

The proposed method is illustrated with swing-up tasks of underactuated inverted pendulums, for which there is extensive literature when the environment is obstacle-free. Under the unconstrained assumption, particular solutions based on energy control [13] and partial feedback linearization [14] are potentially more efficient (in time and control) than a RRT based control. This is due to the use of a discrete set of random actions to expand the RRT, which will almost always be non-optimal choices. However, the same random nature allows the proposed algorithm to find a control solution in a constrained/obstacle-populated workspace, where such particular methods do not apply and where robotics applications, like motion planning and control of a legged robot tend to fall.

III. MOTIVATION AND INTRODUCTION OF THE METHOD

Under differential constraints, a bidirectional RRT (biRRT) works by generating a set of paths, starting from an initial state, each attempting to connect to a goal state. At each iteration, a random state (x_{rand}) is sampled, and the closest node on the forward tree (x_{near}) is identified. Next a series of pre-defined open-loop actions starting from x_{near} generates a set of candidate states. The closest candidate to x_{rand} is selected as a new node x_{new} to be added to the forward tree. The backwards tree is greedily extended in the direction to x_{new}. The trees are then swapped when the number of nodes in each tree is unbalanced. (Details in [15]).

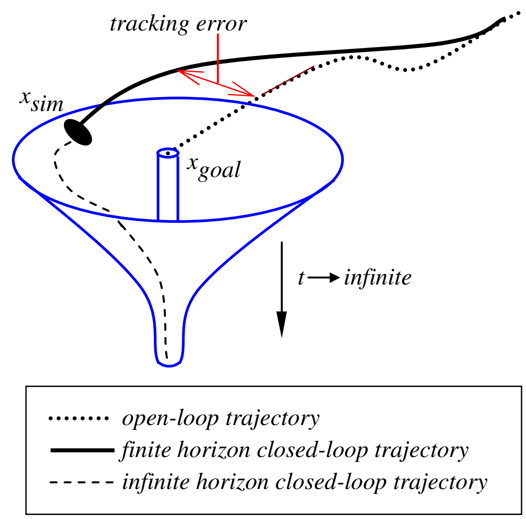


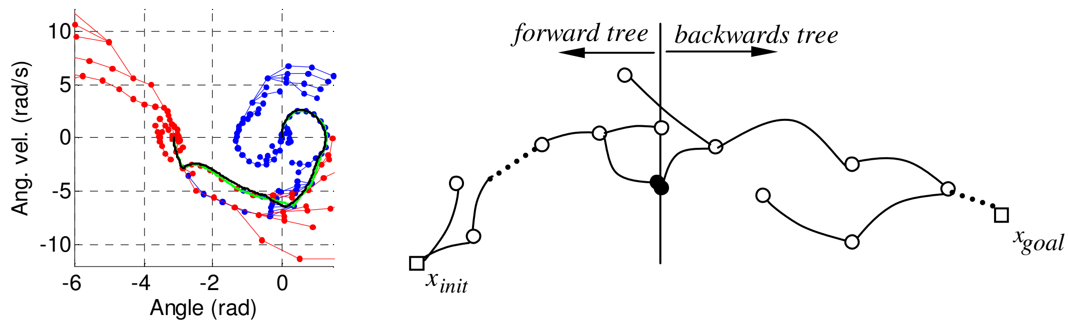
Fig. 1. For purposes of the algorithm propsed, there is no need to track a trajectory perfectly, as long as the last state under a finite horizon controller falls inside the basin of attraction of an infinite horizon controller.

An example of the biRRT used for control is shown in Fig. 2(a). Consider a torque limited pendulum. The figure on the left shows a biRRT search as a phase plot $[\theta \times \dot{\theta}]$ where the state space is defined by $\mathbf{X} = [\theta,\, \dot{\theta}]$. For clarity of explanation, the figure on the right illustrates an equivalent biRRT search drawn with few elements. The goal of this nonlinear control problem is to bring the pendulum from the stable position $\mathbf{X}_{start} = [0,\, 0]^T$ to the upright position $\mathbf{X}_{goal} = [\pm\pi,\, 0]^T$. The limited actuator torque imposes a swing-up action before the pendulum acquires enough momentum to reach the upright position.

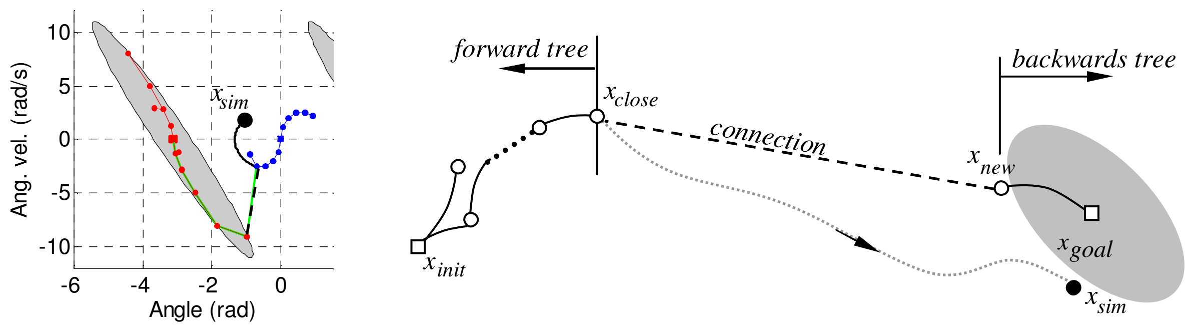
One characteristic of the RRT is that its convergence is solely driven by open-loop actions starting from random initial states. In practice, this means that a solution (if it exists) is sought by the algorithm by continuously sampling until a node in one of the trees is pulled close enough to a node on the other tree, regardless if:

- (1) a certain pair of states (i.e a shortcut connection to a solution) is easily connected by a closed-loop controller,
- (2) connecting nodes do not need to be close, because under feedback control these "jumps" in states can be easily handled by a feedback controller.

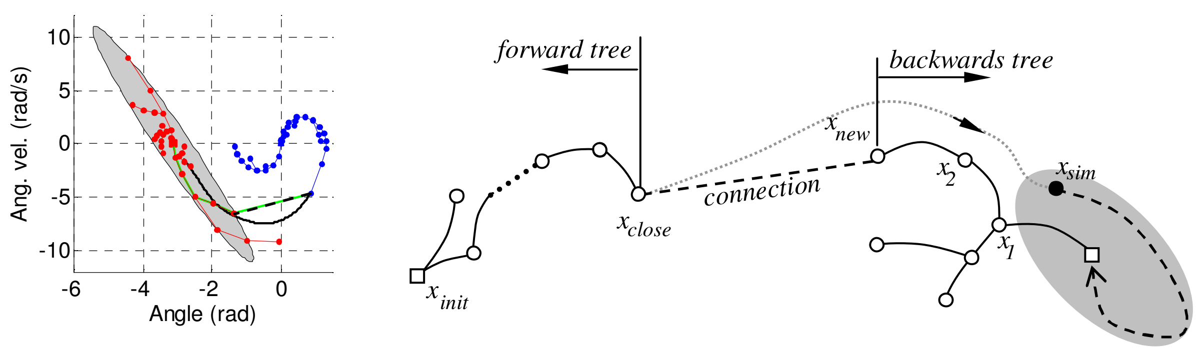
Figs. 2(b)(c) show the proposed feedback approach incorporated to a biRRT. Fig. 2(b) shows an attempt to connect the trees during the first iterations of the biRRT. A local linear feedback controller is used to track the path starting from x_{close}, but because of the nonlinearities and differential constraints involved, the connection between the trees is beyond the robustness of the linear controller and the tracking fails. The "jump" of states is too large and the system states finished at x_{sim}, whereas the ideal tracking (without jumps) should bring it to x_{goal}. As the trees expand in a direction



a) In a conventional biRRT, a solution is found when the trees contain approximately two coincident states, generating smooth transitions between nodes.



b) An attempt to track a distant connection between nodes by feedback control fails. The final state of the finite horizon controller is is out of the stability boundaries of the infinite horizon controller at x_{goal} (gray area).



c) As the biRRT expands, the connection causes less disturbance for the tracking controller. The feedback was able to bring the final state inside the basin of attraction, despite the tracking error.

Fig. 2. While a biRRT must sample until nodes on each tree are close to each other, the proposed algorithm tries to generate and track connections, verifying the feasibility by observing that the final state is stabilizable by the controller at the goal. (Notice that the angles in radians are shown unwrapped).

towards each other, the connection distance decreases, to the point that a closed-loop control succeeds in bringing x_{sim} inside the basin of attraction of an infinite horizon controller designed to keep x_{goal} stable (refer to Fig. 1). This situation is shown in Fig. 2(c), where the total number of nodes (or states) is 63 compared to 202 nodes for the conventional biRRT (Fig. 2(a)).

From a classical control theory perspective, the forced connection in Fig. 2 using a feedback controller is similar to a large trajectory disturbance (or step input) where the reference changes from x_{close} to x_{new}. If the reference change is too large, then the controller will fail to track and eventually destabilize. If the reference change is within the range of the controller action, the RRT search is terminated because the goal is reachable by an infinite horizon stabilizing controller.

A comparison of the exploration required in each case shows the potential of feedback control within a sampled based motion planning structure. Certainly, the very efficient solution of the last example comes at the expense that at each iteration, a local linear feedback controller must be designed and simulated for every iteration of the RRT. While a RRT trajectory can be generated with any kind of forward (black blox) simulator, in the feedback approach, explicit handling of the dynamics (i.e., a model) is required depending on the method used for feedback design and the verification of the basin.

This paper is motivated by the following features of feedback motion planning:

- avoid oversampling of states;
- allow the termination of the RRT exploration as soon as it is made possible by feedback means;
- feedback controller is designed as the RRT expands; and,
- partially optimal trajectories at the neighborhood of the connection are achieved (by using optimal regulators)

IV. THE GAIN SCHEDULED RRT

This section introduces a feedback controller and a verification method for the feasibility of connection of states. This informs the GS-biRRT design.

A. Feedback controller

For systems with a quadratic reward function, optimal full state feedback solutions may be found by solving the Riccati equations to generate LQR gains for both the infinite horizon (via the algebraic Riccati equation) and the finite horizon (via the differential Riccati equation, typically solved by dynamic programming) [2]. These solutions are used for a stabilizing controller at the goal and a tracking controller during the trajectory following phase, respectively.

The use of time variant linear quadratic regulators is suitable in the biRRT framework because the gains can be designed incrementally according to the growth of the backwards tree, as hinted in [6], when the LQR gains are designed as a continuous sequence of gains on each branch of the backwards tree finishing at x_{goal}. In the proposed approach, because the LQR gains must be designed incrementally, for each node, the differential Riccati equation is solved based on the value of the controller on the previous node with:

$$-\dot{P} = PA + A^T P - PBQ_u^{-1} B^T P + Q_x \tag{1}$$

where A, B are the system dynamics linearized at the states of the tree node and Q_x, Q_u are the penalty matrices for state error and control usage, respectively. Feedback control is given by:

$$\delta u(t) = -Q_u^{-1} B^T P(t) x \tag{2}$$

where the $K = Q_u^{-1} B^T P(t)$ is the finite horizon LQR gains. Fig. 3 illustrates the process in which the LQR gains are designed backwards incrementally by integrating Eq. 1 starting with the value of $\mathbf{P}(t_{n-1})$ of the parent node.

For the final expected cost, the value of $\mathbf{P} = \mathbf{P}(t_f)$ is given by the infinite horizon LQR. This not only allows the calculation of the gains at the goal by solving the algebraic Riccati equation [2] in Eq. 3, but also makes it possible to verify the basin of attraction for the time invariant controller (detailed in sec. IV-B).

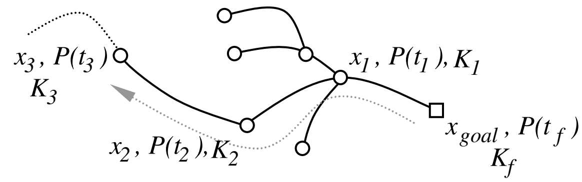


Fig. 3. Incremental generation of LQR gains along the backwards tree (direction represented by the arrow).

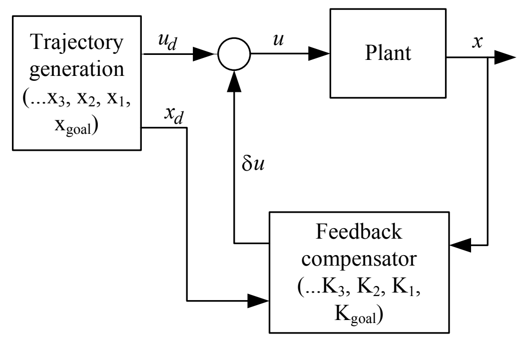


Fig. 4. Two degree of freedom controller. The feedback controller has a sequence of LQR gains scheduled at each tree node that is part of the RRT open-loop trajectory.

$$\dot{P}A + A^T P - PBQ_u^{-1} B^T P + Q_x = 0 \tag{3}$$

For simulating the closed-loop control, a conventional two degree-of-freedom controller [2] is gain scheduled for trajectory tracking shown in Fig. 4. The feedforward compensator (trajectory generation) outputs are defined by an interpolated sequence of states and actions registered in each node of the biRRT. The feedback compensator is scheduled with the interpolated gains of the respective node. The feedback control law is given as:

$$u(t) = K(x - x_d) + u_d \tag{4}$$

where K is the scheduled gain for the corresponding state.

B. Verification of the Connection

Consider again the biRRT in Fig. 2(c). The feasibility of the forced connection between x_{close} in the forward tree with x_{new} in the backwards tree is verified with a forward integration to simulate the closed-loop tracking task. The scheduled controller, drives the states to follow the open-loop trajectory starting from x_{close} and passing through $x_{new}, x_2, x_1, x_{goal}$, respectively; tracked by scheduled controllers designed in section IV-A. Because the connection $x_{close} - x_{near}$ does not address the system dynamics, a tracking error is generated and the final simulated state x_{sim} is not expected to be at x_{goal}. However, if the tracking controller is properly designed, the final integrated state x_{sim} approximates x_{goal} as the greedy biRRT expands and the trees get closer (in a L_2 metric sense).

Lyapunov stability [16] gives the property that if the finite horizon tracking controller brings x_{sim} inside the basin of attraction, then the infinite time horizon controller (Eq. 3) at the goal will drive x_{sim} to x_{goal}. Conversely, if the final state is outside the basin of attraction (shown in Fig. 2(b)), then the forced connection has generated disturbance beyond the tracking ability of the controller, rendering a failed attempt.

Reasoning that a state is inside the basin under a finite horizon control, provides an elegant way of estimating successful connections under a fixed simulation time. This is more efficient than a brute force solution which would consider simulating connection attempt until the system reaches steady-state regime and verifying if it reached x_{goal}.

Estimation of the basin of attraction of the controller (as proposed in [6]) provides a method for the direct computation of Lyapunov functions for both the time invariant LQR (at x_{goal}) and for the states in the vicinity of the open-loop trajectory that falls within the boundaries of the basin at the goal. The method proposed uses the positive definite structure of $V(x) = \mathbf{x}^T P \mathbf{x}$ as a Lyapunov function at the goal (for the local linear system) and find the maximum value of ρ which delimits the boundaries of the stable region:

$$\mathcal{B}_G(\rho) = \{\mathbf{x} | 0 \leq V(\mathbf{x}) \leq \rho\} \tag{5}$$

and the property that $\dot{V}$ is negative definite within the boundaries of $\mathcal{B}$ is verified using convex optimization based on sum-of-squares method (detailed explanation of the method is found in [6], [17]).

C. The Algorithm

BiRRTs are chosen as the basis for implementing the algorithm because their greedy nature that attempts to connect one tree to each other, which makes the exploration shorter. However, the algorithm can be easily reduced to a single forward RRT by simply fixing the backwards tree to a single x_{goal} node with no further adaptation.

The gain scheduled biRRT (GS-biRRT) algorithm is initiated by designing an infinite time LQR at the origin and verifying its basin of attraction, similar to the procedure in [6]. The forward and backwards tree are then expanded as in an ordinary biRRT algorithm.

At every new node x_{new} in the backwards tree, three additional steps are made:

1) A set of finite time LQR gains are calculated based on the previous values of the $\mathbf{P}$ matrix (Eq. 1) recorded in the parent node of x_{new}. The resulting $\mathbf{P}_{x_{new}}$ is then kept with the new node.

2) A direct connection between the closest node x_{close} in the forward tree to x_{new} in the backwards tree is made. Similar to step 1), a set of LQR gains are calculated for x_{close} based on the values of the x_{new} controller (refer again to Fig. 2(c)).

3) Forward simulation connecting x_{close} in the forward tree to x_{new} in the backwards tree while following its parents until x_{goal}. The final state is then checked to see if it is inside the basin.

If the final state is outside the basin, conventional biRRT expansion proceeds. If the final state is inside the basin, the search stops and LQR gains are scheduled by back propagating the solution of Eq. 1 from x_{goal} to x_{init}. The algorithm outputs the sequence of LQR controllers for feedback control, as well as the open-loop trajectory to be tracked.

D. Implementation Details

An alternative to step 3) is to estimate the basins of attraction for each new node on the backwards tree as proposed for the LQR-trees algorithm in [6]. Then it is enough to verify that the feedback controller can drive the states from x_{close} to the basin of x_{new}. However, this requires more computation than a direct forward simulation. Moreover, there is no intention of covering the space with basins of attractions since this work is motivated by single query problems.

In step 3), starting the simulation from x_{close} and not from x_{start} (see Fig. 2(c)) alleviates a full forward integration of the trajectory during the verification of the connection. This can be helpful for long trajectories where simulations are computationally intensive. However, for stability verification of systems with critical nonlinear dynamics, it may be judicious to simulate the full trajectory, especially if the system is stiff and there is concern about numerical error integration during simulation.

The estimation of the basin of attraction does not consider saturation of the actuators. A basin of attraction of a saturated system is smaller than the basin of an unsaturated dynamics, since there is less control authority for stabilization. One way to alleviate this problem is to set a large penalty for the control usage of the LQR controller at the goal ($J = \mathbf{x}^T \mathbf{Q_x} \mathbf{x} + \mathbf{u}^T \mathbf{Q_u} \mathbf{u}$, with $\mathbf{Q_u}$ large). Since a large penalty is not intuitive, tracking robustness of the saturated system is increased by purposely allowing some margin for the feedback control (δu in Fig. 4). The connection is verified with a maximum actuator value that is slightly lower, (e.g. 5 to 25 %) than the real actuator limits.

V. RESULTS AND DISCUSSION

A. Single-Link Pendulum

The GS-biRRT is implemented for the single-link pendulum described in section III. The pendulum has a mass of 5 kg concentrated at the tip, length 0.5 m and damping 0.1 kgm^2/s. The biRRT is expanded and verified with control actions $u = [0\ , \pm 1\ , \pm 2\ , \pm 3]\ Nm$ and the torque limit for

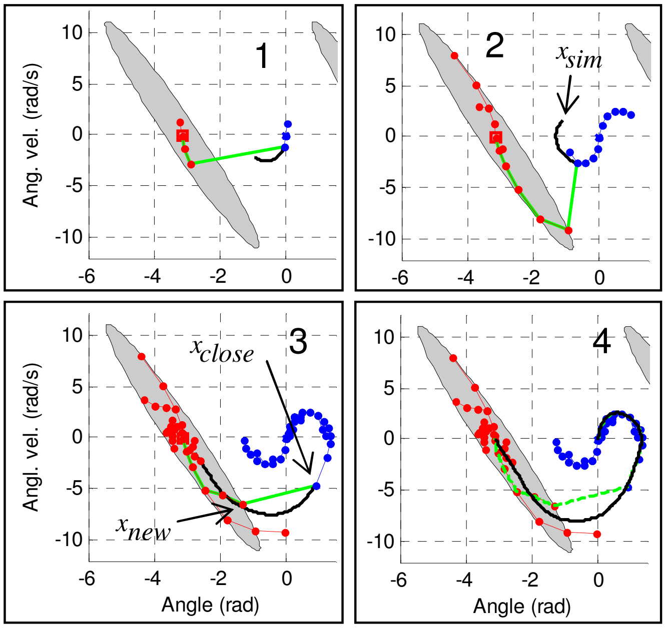


Fig. 5. GS-biRRT expansion. Blue: forward tree, red: backwards tree, black: integrated feedback dynamics, green: forced connected trajectory, gray area: basin of attraction of the infinity horizon controller at the goal. Sequence 1 and 2: connection attempts where the system does not achieve the basin. At step 3 the system reaches the basin. Step 4: complete closed-loop trajectory.

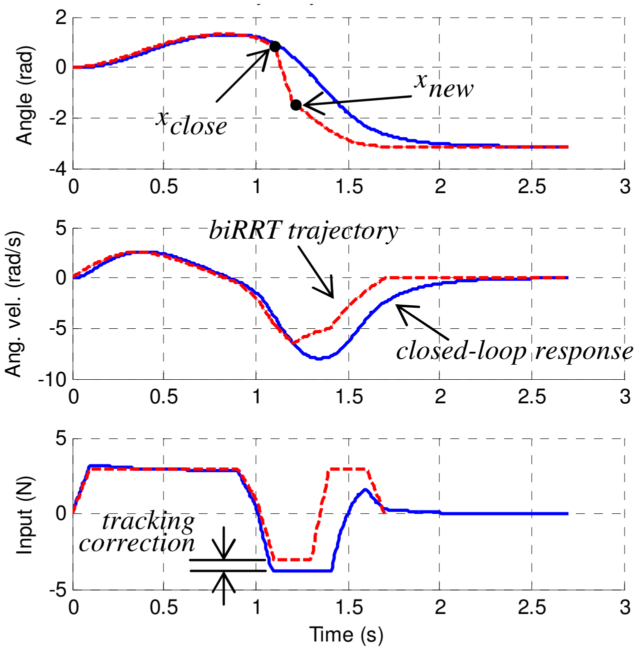


Fig. 6. The forced connection of the GS-biRRT shows as a large trajectory disturbance in time response. A margin for tracking correction is important to afford the discontinuity in trajectory.

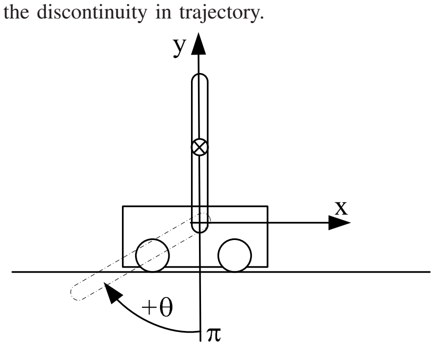


Fig. 7. Cart pole model.

the final feedback controller is 3.75 Nm and 25% of control margin for tracking error corrections.

Fig. 5 shows the result of the proposed GS-biRRT on the phase-plane. In the first two sub figures, the closed-loop controller tries to connect the forward and backwards tree naively without success (note x_{sim} does not reach the basin of attraction). In subfigure 3, x_{sim} finishes inside the basin. Notice that the controller connects the still far x_{close} and x_{near} states where in a normal biRRT approach, random sampling would proceed until these states approximately coincide. Fig. 6 shows the time response of the previous simulation. Notice that forced connection $x_{close} - x_{near}$ creates a large trajectory discontinuity, that requires feedback correction. In an average of ten simulations for each method, the conventional biRRT finds a solution with a tree of 157 ± 78 nodes, while the GS-biRRT needs in average 63 ± 7 nodes (where $\pm\sigma$ is one standard deviation).

B. Cart and Pole Swing-Up

As the dimension of the search increases, the difference in the size of the trees between the biRRT and the GS-biRRT (and, the exploration required to find a solution) becomes more obvious. In part, this is because the linear quadratic regulator is indifferent of the size of the state vector.

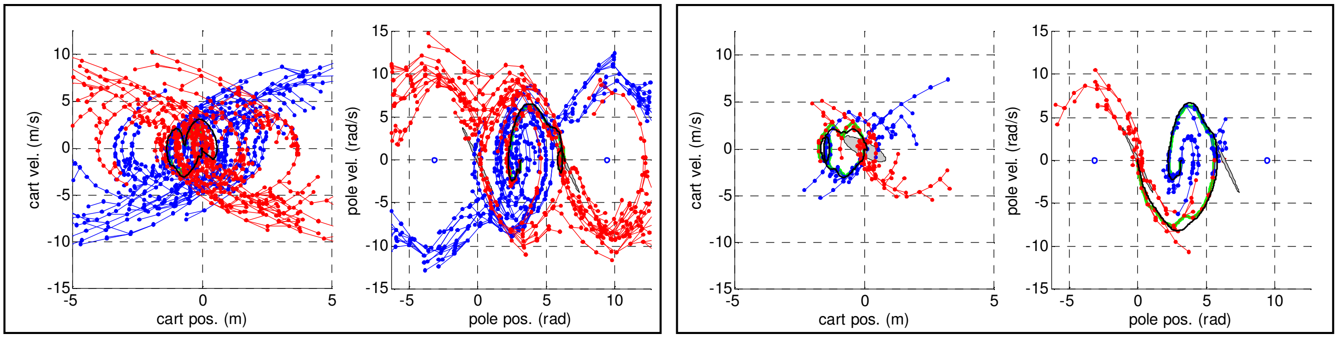
The method is initially applied to swing-up task in an unconstrained workspace (Fig. 7). This canonical nonlinear control theory problem, consists of moving the actuated cart backwards and forwards, so that the unactuated pole is swung-up, from its stable position to the upright position, with the cart resting at its start position.

The state vector is $\mathbf{x} = [x,\, \theta,\, \dot{x},\, \dot{\theta}]^T$, $\mathbf{x}_{init} = [0,\, (2k+1)\pi,\, 0,\, 0]$, $\mathbf{x}_{goal} = [0,\, (2k)\pi,\, 0,\, 0]$, where $k = 0, \pm 1, \pm 2, \pm 3....$ The dynamics are based on the model in [18]:

$$\ddot{\theta}_t = \frac{g \sin\theta_t + \cos\theta_t \left[\frac{-F_t - ml\dot{\theta}_t^2 \sin\theta_t}{m_c + m}\right]}{l\left[\frac{4}{3} - \frac{-\cos^2\theta_t}{m_c + m}\right]} \tag{6}$$

$$\ddot{x}_t = \frac{F_t + ml\left[\dot{\theta}_t^2 \sin\theta_t - \ddot{\theta}_t \cos\theta_t\right]}{m_c + m} \tag{7}$$

where:
$g = -9.8\ m/s^2$, acceleration due to gravity
$m_c = 1.0\ kg$, mass of cart
$m = 0.1\ kg$, mass of pole
$lc = 0.5\ m$, position of center of mass of the pole
$l = 1\ m$, pole length



a) biRRT search finished with 1174 nodes

b) GS-biRRT search finished with 182 nodes

Fig. 8. Swing-up of a cart and pole in a free workspace. Blue: forward tree, red: backwards tree, black: closed-loop response, gray area: basin of attraction of the infinity horizon controller at the goal.

The trees are expanded with the cart forces $u = [0, \pm 2, \pm 4, \pm 6, \pm 8, \pm 10]$ N and closed-loop maximum saturation is 12.5 N. Fig 8 shows the biRRT compared to the GS-biRRT search. The state-space is shown in two separated phase plots for clarity. For an average of ten simulations for each method, the biRRT grew 1660 ± 1221 nodes to find a solution while the GS-biRRT required only 144 ± 85 nodes. Thus, much less exploration was required to find the control sequence. The average execution time of the open-loop swing-up trajectory of the biRRT in the same set of experiments was 4.4 ± 1.0 s and the GS-biRRT 2.0 ± 0.5 s. For the sake of comparison, a solution found by a trajectory optimization routine based on collocation [19] shows that the optimal time swing-up motion is 1.6 seconds.

The use of an RRT framework makes it natural for the GS-biRRT to find control strategies among external constraints (e.g., obstacles) as shown in Fig. 9. In this example, the task is to start with the pole at the $\mathbf{x}_{init} = [0,\ (2k+1)\pi,\ 0,\ 0]$ and finish at $\mathbf{x}_{goal} = [3.7,\ (2k)\pi,\ 0,\ 0]$, while passing under the obstacle in the middle, and while avoiding the rail stoppers.

For comparison, 25 simulations were conducted. For all of them, the GS-biRRT shows a consistent strategy (consisting of accelerating the cart until it passes under the obstacle, decelerating so that the pole goes up, and finally controlling the balance and bringing the cart to the goal position). Unless the greedy element luckily samples close to optimal sequences, the biRRT case is more varied with the cart generally having trouble passing around the obstacle, and again during the final swing-up without hitting the stopper. On average, the length of the curve traced by the poletip (dotted line) for the biRRT and the GS-biRRT case was $11.2 \pm 5.3m$ and $6.0 \pm 0.8m$, respectively. In the latter, the small standard deviation is an indication that all solutions generated are roughly consistent. This is an interesting result because although the algorithm runs over a randomized planner, the feedback controller finishes the search at the first feasible opportunity, which tends to occur when the trees are still simple in shape; and thus, the solutions generate a similar trace in the workspace.

Although the GS-biRRT may not always lead to a faster swing-up trajectories (because the disturbance caused by connecting the trees generates an extra time to balance the pole under feedback), it shows that the biRRT exploration using only random samples leads to unnecessarily long trajectories.

VI. CONCLUSION AND FUTURE WORK

A method to solve nonlinear control problems in constrained workspaces using a feedback motion planning strategy is presented. The RRT framework is used to generate random sampled states, and feedback control is used to connect start and goal states at the first feasible opportunity. The connection is made by attempts in tracking a large disturbance caused by the unnatural connection of distant states. The connections are verified with the use of an estimated basin of attraction. The method avoids oversampling of states and generates feedback gains as part of the process. The optimal control is solved with simple linear LQR controllers, whose design process – different from other dynamic programming strategies – is not affected by the dimension of the problem. While components of this problem have been explored before, no prior work spans the entirety of gain scheduling feedback and motion planning in an integrated manner.

Future effort are looking to verify the feasibility of the GS-RRT method for nonholonomic motion planning and fully actuated mechanisms where feedback linearization may avoid the use of scheduled controllers.

VII. ACKNOWLEDGEMENTS

This work is supported by the Rio Tinto Centre for Mine Automation and the ARC Centre of Excellence program funded by the Australian Research Council (ARC) and the New South Wales State Government.

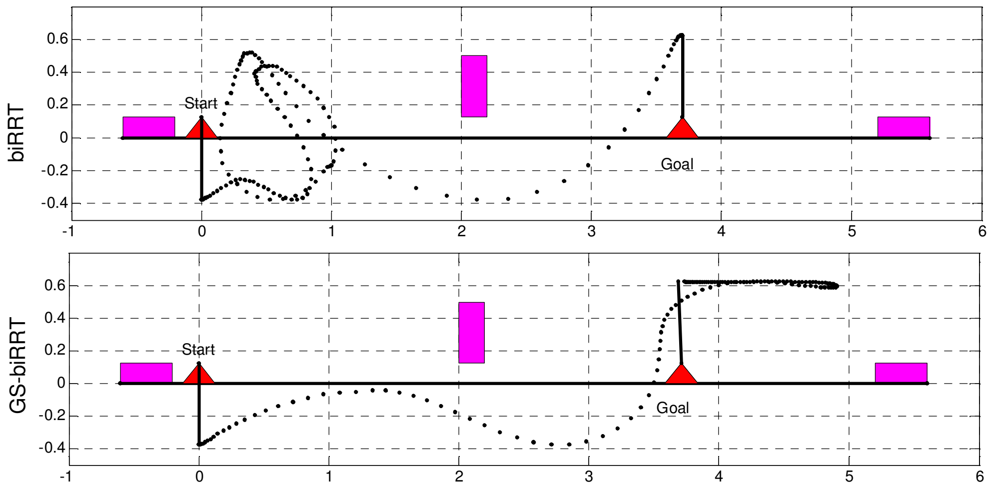


Fig. 9. One of the control strategies for a cart and pole swing-up among obstacles (units in meters). The GS-biRRT shows a motion more natural than the biRRT, consistent in 25 sets of simulation.

REFERENCES

[1] S. LaValle and J. Kuffner Jr, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, p. 378, 2001.

[2] R. M. Murray, *Optimization-Based Control*. Control and Dynamical Systems - California Institute of Technology, 2009.

[3] S. Prajna, P. Parrilo, and A. Rantzer, "Nonlinear control synthesis by convex optimization," *Automatic Control, IEEE Transactions on*, vol. 49, no. 2, pp. 310 – 314, feb. 2004.

[4] A. Yershova, L. Jaillet, T. Siméon, and S. LaValle, "Dynamic-domain rrts: Efficient exploration by controlling the sampling domain," in *IEEE International Conference on Robotics and Automation*, vol. 4. Citeseer, 2005, p. 3856.

[5] M. W. Alexander Shkolnik and R. Tedrake, "Reachability-guided sampling for planning under differential constraints," in *In Proceedings of the IEEE/RAS International Conference on Robotics and Automation (ICRA)*, 2009.

[6] R. Tedrake, "LQR-Trees: Feedback motion planning on sparse randomized trees." in *In Proceedings of Robotics: Science and Systems (RSS)*, 2009, p. 8.

[7] R. Burridge, A. Rizzi, and D. Koditschek, "Sequential composition of dynamically dexterous robot behaviors," *The International Journal of Robotics Research*, vol. 18, no. 6, p. 534, 1999.

[8] M. S. Branicky and M. M. Curtiss, "Nonlinear and hybrid control with rrts," in *Proceedings of the International Symposium on Mathematical Theory of Networks and Systems*, Southbend, IN, 2002.

[9] S. LaValle and J. Kuffner Jr, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, p. 378, 2001.

[10] J. Kuffner and S. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *IEEE International Conference on Robotics and Automation*, vol. 2, 2000, pp. 995–1001.

[11] M. Mason, "The mechanics of manipulation," in *1985 IEEE International Conference on Robotics and Automation. Proceedings*, vol. 2, 1985.

[12] L. Yang and S. Lavalle, "The sampling-based neighborhood graph: An approach to computing and executing feedback motion strategies," vol. 20, no. 3. Citeseer, 2004, pp. 419–432.

[13] K. AsstroKm and K. Furuta, "Swinging up a pendulum by energy control," *Automatica*, vol. 36, pp. 287–295, 2000.

[14] M. Spong, "Partial feedback linearization of underactuated mechanical systems," in *Intelligent Robots and Systems' 94.'Advanced Robotic Systems and the Real World', IROS'94. Proceedings of the IEEE/RSJ/GI International Conference on*, vol. 1, 1994.

[15] S. M. LaValle and J. J. Kuffner, *Rapidly-Exploring Random Trees: Progress and Prospects*, B. R. Donald, K. M. Lynch, and D. Rus, Eds. Wellesley, MA: A K Peters, 2001.

[16] J. Slotine, W. Li *et al.*, *Applied nonlinear control*. Prentice-Hall Englewood Cliffs, NJ, 1991.

[17] R. Tedrake, I. Manchester, M. Tobenkin, and J. Roberts, "LQR-Trees: Feedback motion planning via sums of squares verification," *Under review*, 2010.

[18] A. Barto, R. Sutton, and C. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," vol. 13, no. 5, 1983, pp. 834–846.

[19] P. E. Rutquist and M. M. Edvall, *PROPT - Matlab Optimal Control Software*, Tomlab software, May 2009.