Robust Humanoid Contact Planning with Learned Zero- and One-Step Capturability Prediction

Rushi Bhavesh Shah (rs7236)

I. Summary

The paper describes a new approach to planning the movements of legged robots, such as humanoids, in complex environments. Traditional planners often use quasi-static constraints that limit the possible transitions between different contact points on the ground, but these can quickly lead to planning failure in complex environments. More recently, dynamic feasibility constraints have been proposed, but these still assume fixed and deterministic environments and do not consider the robustness of contact sequences to potential disturbances. The authors of the paper propose a new approach that explicitly takes into account potential disturbances in the environment in order to increase motion robustness. The planner uses neural networks to predict the existence of dynamically feasible capture motions, which can be used to stabilize the robot if a disturbance occurs. This information is then used by the Anytime Non-parametric A* (ANA*) planner to generate footstep sequences that are more robust to disturbances than those generated by conventional distance-based planners. The neural networks are trained using data generated offline with a kino-dynamic optimizer, which allows them to predict the robot's capturability under disturbances. The networks can predict both zero-step and one-step capturability for a full-body dynamic model using both foot and palm capture motions. This allows the planner to generate robust footstep sequences that can adapt to changes in the environment and maintain stability even in the face of disturbances. Humanoid footstep planning is a field of research that aims to develop algorithms and techniques for humanoid robots to plan and execute footstep sequences in various environments. Conventional approaches often rely on avoiding obstacles and assume a flat or piecewise-flat ground, but these approaches may not be effective in more complex environments where dynamic feasibility must also be considered. Dynamic feasibility refers to the ability of a robot to maintain balance and stability while executing a given motion. In these environments, the robot must be able to adapt to changing terrain and external disturbances, such as pushes or slips, in order to successfully navigate the environment and achieve its goals.

To address the limitations of conventional approaches, some recent research has used quasi-static balance criteria to verify dynamic feasibility. This approach involves analyzing the robot's static stability in different configurations and determining whether a given motion is dynamically feasible based on the robot's ability to maintain balance. However, this approach can be overly conservative and may not consider all possible motions.

Other work has combined contact planning with dynamics optimization to address dynamic feasibility in complex environments. This approach involves planning the locations and timing of the robot's contacts with the ground, as well as optimizing the motion of the robot's body and limbs to achieve a desired goal. While this approach can be effective, it can also be computationally expensive and inefficient in complex environments.

In this work, the authors propose a neural network-based approach for generating footstep sequences for a humanoid robot that maximizes the probability of the robot reaching a goal region without falling due to external disturbances. The planning process takes into account the robot's initial stance, the environment, and potential disturbances, and allows the robot to use both its feet and hands to reject disturbances. The friction coefficient and timing of contact transitions are also considered. The authors demonstrate the effectiveness of their approach through experiments in various challenging scenarios, showing that their neural networks achieve high accuracy in predicting robot capturability and that their planning approach generates footstep sequences that are more robust to external disturbances than conventional distance-based planners. This represents a significant advancement in the development of legged robots that can effectively navigate complex environments while maintaining stability under disturbances.

The kino-dynamic optimization method is a technique used for computing dynamically-consistent whole-body motions for a robot. It involves decomposing the problem into two parts: a dynamic optimization problem based on the centroidal dynamics of the robot, and a kinematic optimization problem for the full-body motion. The algorithm iteratively solves both problems until they reach consensus on the center of mass, linear, and angular momentum trajectories, resulting in a locally optimal solution to the original problem. The dynamic optimization problem is solved using a fixed-time algorithm, and the solution is used to compute dynamically robust motions for the robot.

In this method, the authors do not consider collision avoidance constraints in the optimization, but plan to incorporate these in future work. A contact transition is considered capturable if the algorithm converges to a solution that satisfies all constraints after a maximum number of iterations. The authors assume that the probability distribution of external disturbances that affect the linear centroidal momentum of an object is known and fixed, and that they can be represented by a set of samples taken from the distribution. The probability of each sample occurring within a given time period is calculated by integrating the probability over the Voronoi cell of the sample. The authors also assume that only one disturbance will occur within a short time period.

The method includes a method for evaluating the capturability of a robot, either by making new contacts or without making new contacts. The initial state of the optimization includes the robot's centroidal dynamics state immediately after a disturbance, as well as the existing contact poses. For one-step capture, a target contact pose is also specified. The optimization is run for three iterations, and if a kino-dynamically feasible solution is found such that the linear and angular momentum converge to zero at the end of the motion, then the robot is considered capturable.

To reduce computational complexity, the method involves training a set of neural network classifiers offline to predict whether the robot can capture itself under a set of disturbances. The classifiers take the initial standing foot pose, the capture contact pose, and the initial linear and angular velocities as inputs and output a binary value indicating whether the optimizer can find a kino-dynamically feasible solution to capture the robot. This allows the classifiers to focus on poses that are feasible and useful for the robot's intended application.

The Anytime Nonparametric A* (ANA*) algorithm is a variation of the A* algorithm that can be used to solve the contact planning problem for robots. It is designed to quickly find a feasible solution to the problem, and then improve that solution over time. The cost of each action in the ANA* algorithm is defined as a function of the distance between the initial and final states, as well as a term that takes into account the robot's ability to capture itself (i.e. maintain its balance) during the transition. This capture probability can be calculated using either a discretization of the swing phase of the contact transition or a worst-case estimate of the robot's center of mass position and linear momentum. The researchers evaluated the proposed approaches for path planning for robots in three test environments in simulation, comparing them with a baseline approach. They set various parameters, including the friction coefficient, which was set to 0.5. They only considered the rare but dangerous case where high disturbances acted on the robot, and computed the probability that the robot finished the path without falling to evaluate the path quality.

II. Course Relevance

During the course, while learning various path planning algorithms, we learnt that Shortest path algorithms are a type of graph search algorithm that are used to find the shortest path between two nodes in a graph. They can be used to solve a variety of problems, such as finding the shortest route between two locations on a map or routing traffic on a network. There are several different algorithms for finding the shortest path, including Dijkstra's algorithm, A* search, and Bellman-Ford algorithm.

Dijkstra's algorithm is a popular choice for finding the shortest path in a graph because it is efficient and easy to implement. It works by starting at the source node and exploring all of its neighbors, keeping track of the shortest known distance to each node. As the algorithm progresses, it continues to explore the neighbors of each node, updating the distances as necessary until it reaches the destination node. However, Dijkstra's algorithm only works for graphs with non-negative edge weights.

A* search is another popular shortest path algorithm that combines the benefits of Dijkstra's algorithm with the use of a heuristic function to guide the search. The heuristic function estimates the distance from the current node to the destination node, allowing the algorithm to prioritize nodes that are likely to be closer to the destination. This makes A* search more efficient at finding the shortest path than Dijkstra's algorithm.

The Bellman-Ford algorithm can handle graphs with negative edge weights, but it is slower than Dijkstra's algorithm and A* search. It works by iteratively relaxing the distances to each node.

ANA*, an extension of A*, uses a nonparametric model to estimate the cost of a path, rather than using a fixed heuristic function. This allows ANA* to adapt to the structure of the search space as the search progresses, making it more efficient and effective at finding solutions. Like A*, ANA* uses a priority queue to prioritize the search for nodes that are likely to be on the optimal path, and it maintains a set of closed and open nodes to keep track of which nodes have been already searched and which are still being considered.

III. Discussion

It sounds like the authors of the paper have developed a machine learning approach that uses neural networks to predict the probability of successfully rejecting a disturbance at each time step, and an optimization algorithm to find a sequence of footstep locations that maximizes this probability. While this approach can be efficient, there may still be time steps where the robot fails to reject the disturbance. To improve the quality of the solution, the authors suggest that the center of mass position and centroidal momentum could be included as decision variables in the planner. However, this would increase the time complexity of the planning process. It may be possible to find a balance between the accuracy of the solution and the time complexity of the planning process by carefully selecting the variables that are included in the planner and using appropriate optimization techniques. It could be interesting to see the authors' learning approach used in conjunction with a real-time controller for selecting the next contact location in order to stabilize the robot. This could potentially improve the ability of the robot to reject disturbances in real-time and improve overall stability. It would be interesting to see how this approach compares to other methods of disturbance rejection and stability control in terms of performance and efficiency.

REFERENCES

[1] Lin, Yu-Chi, Ludovic Righetti, and Dmitry Berenson. "Robust humanoid contact planning with learned zero-and one-step capturability prediction." IEEE Robotics and Automation Letters 5.2 (2020): 2451-2458.