Applications of Sample Based Model Predictive Control in Industrial Robotic Settings

Noah Harvey, Joseph Burley

ok, but How?

Abstract—**Today many Industrial systems are controlled using feedback-loop based controls; commonly, the PID and state-space controllers. There are draw-backs however as PID controllers tend to be difficult and error prone in tuning and optimizing. Novel approaches to control systems have been made in recent years which ~~incorporate~~ sample-based path finding algorithms (typically the A*) with model predictive controllers (MPC). We discuss the applications of sample based MPCs in industrial robotic applications.**

?

A*

passive voice or impersonal? i.e. It is discussed

I. Introduction

Most industrial robotic applications today involve repetative motion tasks, with little change in the operations of the robots themselves. Typically they are programmed to optimally perform a single task. However today, with the growing rates of production and ranging environment types we find that robots are needed in more dynamic environments. For example Amazon has begun seeking robots that can perform item selection in warehouses [1] which requires the robots to be aware of robots surrounded in a changing environment. Consider also the cost of programming robots to perform very specific tasks. For industrial settings where the product may change from day to day - such as glass manufacturing - it is difficult to constantly reprogram and calibrate a robot to conform its kinematic path planning algorithms to suite environment or product constraints.

?

Adding an example could be better about it.

Model predictive controllers (or MPCs) boast two solutions to the changing environment and constraints problem. First, MPC algorithms allow us to use kinematic models of the robot into the problem directly - this is more a legacy system support to older control methods. Second, and more importantly, we can incorporate system (both robotic and environmental) constraints on our control model. According to [2] MPC has been mostly restricted to slower responding systems such as chemical manufacturing facilities. Recent approaches to autonomous vehicle applications have integrated MPC algorithms as a means of path planning [3], [4] and used improved optimization stage algorithms (we will explain this further in section §II) for generating flexible and optimal controller outputs. This optimization algorithm is known as sample-based model controller (SBMPC).

Interactions with virtual reality?

II. Model Predictive Controllers

Before we discuss SBMPC we note to the reader that because the scope of this paper is non-technical we do not wish to show the gritty details of the SBMPC algorithm. Instead we will do a review of the basic concepts.

this saying positive way

The basic MPC algorithm is as follows[2]:

1) Obtain the current state of the system being controlled.
2) Determine the optimal control inputs ($u_{t+1,t+2,\ldots,t+N}$) by solving an optimization problem that compares the predicted system output with a desired system output while satisfying system constraints.
3) Send the first generated control input u_{t+1} to the plant, update the system state variable and then repeat the process until the system has reached the desired goal or state (depending on the application).

This algorithm can be explained through an example. Suppose that we wish to fire a rifle at a moving target. We could generate a kinematic model for our system and generate some constraints (such as the bullet must be restricted to

→ My recommendation is to add a definition of MPC and some algorithms examples. Some figures to understand the approach could be ideal; rifle example is fine but you can improve with more examples

→ What other researchers (recently) are working on MPC ... what is important to explore and why?

Nice rifle example but could be better related to the 3 points explained before. A picture explaining could be good

a certain trajectory to avoid damaging property or humans). Our system inputs would be the angle of the rifle both vertically and horizontally. The state could be the kinetic and kinematic properties of the bullet and target. The system output would be the location of the bullet relative to the target.

For the sake of the example let us assume that we can freeze the system at anytime wherein our clock will also stop. To get hit on the target we first, freeze our system. Then we calculate an optimal set of system inputs for the next N time steps while obeying our system constraints. Once an optimal predicted trajectory is determined we then implement our first set of these system inputs while unfreezing our system. Practically, in industry, the system is actually never "frozen" but the prediction step is performed fast enough that the system is assumed to not have changed much in its state.

III. Sample Based Model Predictive Controllers

We now explain the "sample-based" part of SBMPC. It was discovered that sampling-based motion planning algorithms are great for solving basic optimization problems. Such algorithms include rapidly exploring random trees, and A* algorithms. Both algorithms were designed for sampling some search-space and generating a graph that contained an optimal path for some start and goal nodes. For MPC we use these methods for the optimization step which yield faster path planning times, and more reliable path planning results.

?

The novel algorithm is based on the A* algorithm where the nodes are control state configurations and edges are the system inputs to transform the system from one control state to the next. This sampling optimization algorithm as discussed in [4] is as follows:

1) Generate a set of samples of the control space.
2) Use the generated control space samples to generate state-space samples.
3) Use an A*like heuristic to evaluate the cost of each node based on the desired objective.
4) Iterate through the nodes (starting with the lowest cost node first) and compute the edge cost for that node starting from the current node. Stop when an edge cost is evaluated that fits the system constraints.
5) Continue the process until the system has reached its goal.

IV. Applications of SBMPC

From [2], [4], [3] we have three robotic applications for SBMPC. All three are applied to path planning for mobile robots. Within such applications the control variables are typically kinematic based such as position and velocity. Current applications include generating optimized paths to a goal for mobile robots. According to [3] SBMPC is also suitable for path planning autonomous underwater vehicles in searching out mines. In [4] we find that SBMPC is good for detecting and avoiding local minimum in the system state and perform path planning accordingly.

Ok you are mentioned other people works but you need to be more specific

Industrial robotic applications are numerous. One import application of SBMPC is object avoidance of dynamic environments - such as with other moving machinery interacting with the robots. The integration point for SBMPC algorithm would be for online path planning of the robot. The goal being of course to move the robot in such a way to avoid detected objects in the workspace. For example a kinematic model of a 3 prismatic joint robot we have the following kinematic model:

name some that industry wants to know and why do you have $$ impact safety issues?

$$\vec{x}_{k+1} = A\vec{x}_k + B\vec{u}_k \tag{1}$$

$$\vec{y}_{k+1} = C\vec{x}_k + \vec{d} \tag{2}$$

where A,B,C form the state space model for the system. $\vec{x}$ is the state of our system, $\vec{u}$ is the joint space acceleration, and $\vec{y}$ is the position and velocities of the end-effector.

Along with the state space model the constraints of our system will be:

$$\vec{u}_l \leq \vec{u}_k \leq \vec{u}_u \tag{3}$$

$$C(\vec{y}) \leq 0 \tag{4}$$

where $C(\vec{y})$ are the collision constraints on the end-effector. The end-effector. Our fitness functions would consider the least amount of additional acceleration required to move the end-effector towards a target while avoiding known obstacles.

In section III ... similar section II ...
Improve definition of SBMPC ... why is important to explore them?
What other researchers are doing related to you, ...
what they found? why is important to continue in the way you are proposing.?

V. Conclusion

Model predictive control has been an important step in controlling both linear and nonlinear systems with internal and environmental constraints. The detriment of MPC is that its computation time is rather slow and the prediction tuning can become rather complicated. Sample-based MPC are a novel algorithm for use in industrial and mobile robotic settings that optimizes the prediction step of the MPC algorithm. The benefit of SBMPC algorithms - aside from optimized prediction - is that by sampling the control and input space more constraints can be handled by the system with the same manner of efficiency. A proposed application is the use of SBMPC algorithms to perform kinematic control of serial robots for simultaneous path planning and object avoidance.

could be nice if you show ~~show~~ in your essay the benefits of SBMPC with some clear examples.

References

[1] L. et. al., "Automated robot picking system for e-commerce fulfillment warehouse application," *The 14th IFToMM World Congress*, 2015.

[2] D. et. al., "Sampling based model predictive control with application to autonomous vehicle guidance," *Conference on Recent Advances in Robotics*, 2008.

[3] C. et. al., "Application of sampling based model predictive control to an autonomous underwater vehicles," *Ship Science and Technology*, vol. 4, 2010.

[4] D. et. al., "Motion planning for steep hill climbing," *IEEE International Conference on Robotics and Automation*, 2011.

Good paper structure!