

АЛГОРИТМ УПРАВЛЕНИЯ ЧЕТВЕРОНОГИМ РОБОТОМ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ MPC И SQP

М.В. Ляховский

Университет ИТМО

Россия, 197101, Санкт-Петербург, Кронверкский проспект, 49
E-mail: 337852@niuitmo.ru

Д.Н. Базылев

Университет ИТМО

Россия, 197101, Санкт-Петербург, Кронверкский проспект, 49
E-mail: bazylevd@itmo.ru

Ключевые слова: оптимальное управление, четвероногий шагающий робот, управление с прогнозирующими моделями, последовательное квадратичное программирование.

Аннотация: В работе рассматривается задача управления походкой четвероногого робота с использованием управления с прогнозирующими моделями. Синтезирован регуляризованный предиктивный регулятор с применением эвристических функций для упрощения сложных целевых функций и быстрого поиска осуществимых решений задачи оптимального управления. Предложенный подход позволяет улучшить производительность основного цикла системы управления четвероногим шагающим роботом за счет решения нелинейной задачи оптимизации с помощью метода последовательного квадратичного программирования.

1. Введение

Методы управления четвероногими роботами на основе модельно предиктивного регулятора (Model Predictive Control – MPC) позволяют достичь устойчивого передвижения для большого количества походок на различных скоростях [1]. В работе [2] предложен регуляризованный предиктивный регулятор (Regularized Predictive Control – RPC) с использованием эвристических функций, которые позволяют аппроксимировать сложную динамику и учитывать ошибки, возникающие из-за упрощений модели и неопределенности параметров. Решаемая задача оптимизации в RPC является нелинейной, что приводит к большому количеству способов нахождения минимума целевой функции [3]. Для решения нелинейной задачи оптимизации предлагается использовать метод последовательного решения подзадач квадратичного программирования – SQP [4].

2. Описание алгоритма управления

2.1. Регуляризованный предиктивный регулятор

В качестве модели четвероного шагающего робота выбрана модель абсолютно твердого тела. Поскольку масса ног робота, зачастую, составляет малую часть массы

тела, то движение ног незначительно влияет на движение тела робота в целом. В связи с этим, массой ног можно пренебречь. Состояние системы в следующий момент времени рассчитывается на основе текущего состояния робота, его модели и заданного управления. Сформулируем задачу оптимального управления как задачу по нахождению минимального значения скалярной целевой функции. Одним из наиболее простых способов задания целевой функции является квадратичный вид функции. Данная функция является гладкой, дважды дифференцируемой и имеет один минимум. Целевая функция строится на основе отклонения текущего состояния системы от желаемого и значения управления [5].

В методе RPC используются эвристические функции, которые позволяют упростить сложные целевые функции и выполнить быстрый поиск осуществимых решений. В результате синтезируется управление, которое в значительной степени полагается на эвристики, а не на точную модель или реалистичную динамику. Подобный подход помогает найти оптимальное решение без использования сложных вычислений и при этом не оказывает негативное влияние на поиск возможного лучшего решения в окружающем пространстве целевой функции.

Эвристики для переменных оптимизации добавляются в исходную целевую функцию через ошибку состояния:

$$\begin{aligned} (1) \quad & \tilde{X} = H_X(X, \Phi, x_d) - X, \\ (2) \quad & X = [x^T u^T]^T, \\ (3) \quad & x = [p^T \theta^T \dot{p}^T \dot{\theta}^T]^T, \\ (4) \quad & u = [r_1^T f_1^T \dots r_4^T f_4^T]^T, \end{aligned}$$

где x – вектор состояния, p – положение центра масс робота относительно мировой СК, θ – ориентация тела, выраженная углами Эйлера в порядке ZYX, \dot{p} – линейная скорость тела робота в мировой СК, $\dot{\theta}$ – производная углов Эйлера по времени; u – вектор управления, r_i – вектор между центром масс робота и точкой контакта i -ой ноги с землей, f_i – вектор силы, приложенной i -ой ногой в точке контакта с землей; X – вектор переменных оптимизации, состоящий из комбинации вектора состояния и вектора управления; $H_X(X, \Phi, x_d)$ – эвристическая функция, Φ – фаза походки, x_d – желаемый вектор состояния.

Задача оптимального управления:

$$\begin{aligned} (5) \quad & \min_X J(X) = \sum_{k=0}^{N-1} \tilde{X}_k^T W_k \tilde{X}_k \\ & s. t. x_{k+1} = A_k x_k + B_k h(X_k, \Phi_k) + d_k, \\ & g(X_k, \Phi_k) \leq 0 \end{aligned}$$

$$(6) \quad A_k = \begin{bmatrix} I_6 & dt_k I_6 \\ 0_6 & I_6 \end{bmatrix} B_k = \begin{bmatrix} \frac{dt_k^2}{2} I^{-1} \\ dt_k I^{-1} \end{bmatrix} d_k = \begin{bmatrix} \frac{dt_k^2}{2} a_g \\ dt_k a_g \end{bmatrix},$$

$$(7) \quad h(X_k, \Phi_k) = \begin{bmatrix} f_B \\ f_T \end{bmatrix} = \sum_{i=1}^4 \begin{bmatrix} I_3 \\ R(\Theta_k)^T [r_{i,k}] \end{bmatrix} S_i,$$

$$(8) \quad a_g = [g^T 0_{3 \times 1}^T]^T,$$

где $J(X)$ – целевая функция, W_k – матрица весов, A_k, B_k, d_k – матрицы нелинейной модели на шаге k , I – диагональная единичная матрица, dt – шаг дискретизации, R – матрица поворота, g – вектор гравитации, $g(X, \Phi)$ – физически реализуемые ограничения системы в виде неравенств, S_i – матрица выбора ног в фазе контакта.

На рис. 1 приведен пример, позволяющий продемонстрировать эффекты регуляризации. Зададим исходную целевую функцию J в качестве нелинейной функции с несколькими локальными минимумами. В результате сложения квадратичной функции регуляризации с исходной функцией получается комбинированная целевая функция, потенциально с единственным минимумом. Таким образом, при

регуляризации форма новой целевой функции изменяется с учетом исходной нелинейной функции.

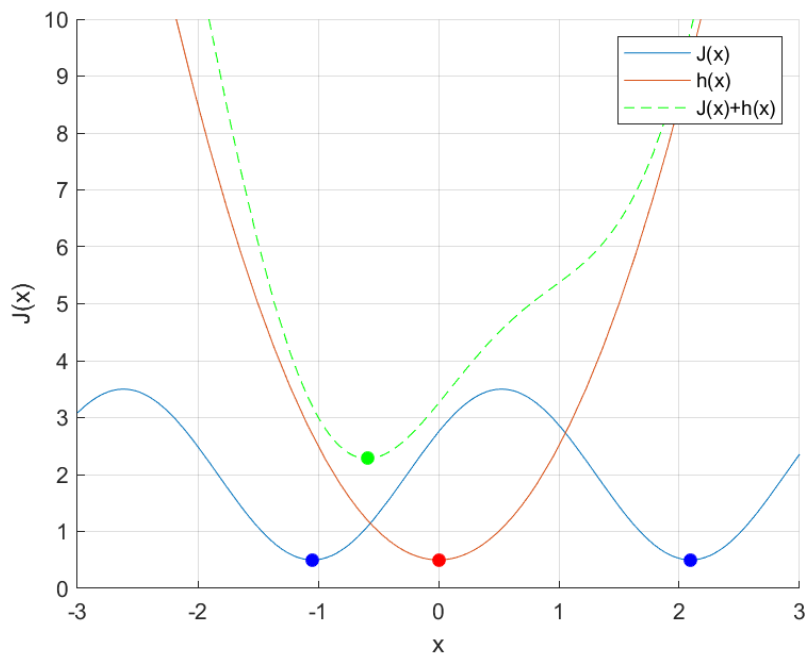


Рис. 1. Демонстрация влияния эвристики на исходную нелинейную функцию с множеством минимумов.

Несмотря на преимущества использования эвристических функций в подходе RPC, их разработка на практике является нетривиальной задачей и может сопровождаться рядом затруднений. Заметим, что в общем случае, непосредственное добавление функции регуляризации не упрощает задачу оптимизации. Плохо спроектированная функция регуляризации может отодвинуть комбинированный минимум от лежащего в основе минимума, что может привести к генерированию некорректных управляющих воздействий, ухудшению качества походки и даже потере устойчивости робота. В то же время достаточно хорошо спроектированная эвристика позволяет существенно упростить окружающее пространство портрета целевой функции, не оказывая влияния на оптимальное решение.

2.2. Последовательное квадратичное программирование SQP

Последовательное квадратичное программирование (Sequential Quadratic Programming – SQP) – один из наиболее распространённых и эффективных оптимизационных алгоритмов общего назначения. Основная идея данного метода заключается в последовательном решении задач квадратичного программирования, аппроксимирующих исходную задачу оптимизации.

Составим Лагранжиан нелинейной задачи оптимизации (Non Linear Programming – NLP):

$$(9) \quad L(w, \lambda_g) = \phi(w) + \lambda_g^T g(w),$$

где w – вектор переменных оптимизации, ϕ – целевая функция, λ – множитель Лагранжа, g – ограничения системы.

Далее, выполним разложение Лагранжиана в ряд Тейлора до второго порядка. В результате, сформулируем задачу квадратичного программирования (Quadratic Programming – QP):

$$(10) \quad \begin{aligned} & \min_{dw} \nabla_w \phi(w_i)^T dw + \frac{1}{2} dw^T B_i dw, \\ & s. t. g(w_i) + \nabla_w g(w_i)^T dw = 0, \\ & dw = w - w_i \end{aligned}$$

где dw – шаг решения, B – матрица Хесса от Лагранжиана.

Рассмотренная задача QR будет выпуклой если матрица Хесса будет положительно полуопределенной. Для того чтобы гарантировать это свойство нужно аппроксимировать исходную матрицу Хесса. Один из наиболее простых способов аппроксимации может быть задан в виде:

$$(11) \quad H = J^T J,$$

где H – аппроксимация матрицы Хесса, J – якобиан.

Так как RPC вычисляет управление на некоторый горизонт, последующие вычисления RPC достаточно схожи и могут быть инициализированы решением, полученным на прошлой итерации со смещением на один шаг назад. В таком случае финальный шаг станет неизвестным, но его можно инициализировать значением с предыдущего шага. Кроме того, критически важно находить решения в режиме реального времени, так как система управления будет использоваться на борту робота. Поэтому на каждой итерации работы RPC решение задачи оптимизации выполняется только за один шаг SQP. Таким образом решение улучшается со временем, а не повторяется до сходимости для каждой итерации.

3. Заключение

В работе рассмотрена задача оптимального управления походкой четвероногого робота с использованием регуляризованного предиктивного регулятора. Одно из ключевых достоинств подхода RPC заключается в том, что данный алгоритм позволяет внедрять в задачу оптимизации достаточно богатую информацию о роботе в виде эвристических функций без внесения изменений в структуру системы управления. Преимуществом использования метода последовательного квадратичного программирования для решения нелинейной задачи оптимизации является упрощение расчетов в момент вычисления решения. Также метод SQP позволяет с течением времени получить сходимость решения к физически реализуемому.

Список литературы

1. Carlo J., Wensing P., Katz B., Bledt G., Kim, S. Dynamic locomotion in the MIT cheetah 3 through convex model-predictive control // 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2018. P. 1-9. DOI: 10.1109/IROS.2018.8594448.
2. Bledt G., Kim S. Extracting legged locomotion heuristics with regularized predictive control // 2020 IEEE International Conference on Robotics and Automation (ICRA). 2020. P. 406-412. DOI: 10.1109/ICRA40945.2020.9197488.
3. Bledt G., Wensing P.M., Kim, S. Policy regularized model predictive control to stabilize diverse quadrupedal gaits for the MIT cheetah // 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2017. P. 4102-4109. DOI: 10.1109/IROS.2017.8206268.
4. Grandia R., Hutter M., Jenelten F. Perceptive Locomotion through Nonlinear Model Predictive Control. 2022. DOI:10.48550/arXiv.2208.08373.
5. Bellicoso C.D., Jenelten F., Gehring C., Hutter M. Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots. IEEE Robotics and Automation Letters. 2018. Vol.: 3, No. 3. P. 2261-2268. DOI: 10.1109/LRA.2018.2794620.