

18. Logunov A.V., Konn V.Yu., Beresnev A.L. Diagnostika avtomobil'nogo dvigatelya vnutrennego sgoraniya po vibroakusticheskim signalam [Diagnosis of a car internal combustion engine by vibroacoustic signals], *Nauchnye issledovaniya i razrabotki poslednego desyatiletiya vzaimodeystvie proshlogo i sovremennogo: Sb. statey* [Scientific research and development of the past and present decade interaction: collected articles]. Rostov-on-Don, 2019, pp. 77-80.
19. Logunov A.V., Beresnev A.L. Vibroakusticheskaya diagnostika transportnykh sredstv [Vibroacoustic diagnostics of vehicles], *Problemy sovremennoy sistemotekhniki: Sb. nauchnykh statey* [Problems of modern system engineering collection of scientific papers]. Issue XIV. Taganrog, 2020, pp. 118.
20. Logunov A.V., Konn V.Yu., Beresnev A.L. Vibroakusticheskaya diagnostika avtomobiley [Vibroacoustic diagnostics of cars], *Prikladnye informatsionnye sistemy v tekhnologiyakh nazemnogo transporta (mashinostroenie): Mater. II Vserossiyskoy nauchno-prakticheskoy konferentsii s mezhdunarodnym uchastiem* [Applied information systems in land transport technologies (mechanical engineering). Proceedings of the II All-Russian Scientific and Practical Conference with international participation]. Taganrog, 2020, pp. 65.

Статью рекомендовал к опубликованию к.т.н. А.И. Дордопуло.

Логунов Артём Владимирович – Южный федеральный университет; e-mail: artem-logunov@yandex.ru; г. Таганрог, Россия; тел.: +79054586468; аспирант.

Береснев Алексей Леонидович – e-mail: alex_tr6@mail.ru; кафедра электротехники и механики; к.т.н.; доцент.

Logunov Artem Vladimirovich – Southern Federal University; e-mail: artem-logunov@yandex.ru; Taganrog, Russia; phone: +79054586468; postgraduate student.

Beresnev Alexey Leonidovich – e-mail: alex_tr6@mail.ru; the departments electrical engineering and mechatronics; cand. of eng. sc.; associate professor.

УДК 519:71/ 519.7

DOI 10.18522/2311-3103-2022-1-172-182

А.А. Ткаченко, Д.Д. Девяткин

ПРОЕКТИРОВАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ И ОБЕСПЕЧЕНИЕ АВТОНОМНОСТИ ДЛЯ МОБИЛЬНОГО ДВУХКОЛЕСНОГО РОБОТА

Управление с прогнозированием (Model Predictive Control) – это усовершенствованный метод управления процессами, который используется при соблюдении набора ограничений. С инженерной точки зрения MPC-метод проектирования систем управления является привлекательным, т.к. является сравнительно простым при проектировании, в том числе для решения сложных производственных задач. Данный метод схож с классическим синтезом системы управления на основе линейно-квадратичного регулятора (LQR). Ключевое различие между MPC и LQR заключается в том, что управление с прогнозированием решает задачу оптимизации в пределах скользящего временного горизонта, в то время как линейно-квадратичный метод используется для решения той же задачи фиксированное временное окно. В работе рассматривается способ построения системы управления для двухколесного мобильного робота с использованием Model Predictive Control. Приведен процесс построения математической модели механической системы робота, а также выполнена линеаризация полученной модели. Представлены основные принципы построения системы управления на основе MPC для линейных систем без внешних возмущений, а также с использованием наблюдателя для оценки состояния модели при влиянии аддитивных белых гауссовских шумов. Рассмотрен вариант синтеза системы управления с накладываемыми ограничениями на входной сигнал. Также представлен способ определения положения двухколесного робота в пространстве с помощью системы технического зрения, которая основана на использовании искусственной нейронной сети. Приведена архитектура модели, использующаяся совместно со стереокамерой, с помощью которой реализуется построение карты глубины изображения. В качестве архитектуры нейронной сети используется,

зарекомендовавшая себя модель YOLOv3. В работе описываются модели, которые способны проводить обработку данных в режиме реального времени. Помимо вышеперечисленного в работе подробно описывается принцип работы модели глубокого обучения – YOLOv3, которая основана на нескольких блоках обработки входных данных. Представлено подробное описание реализации стереокамеры в связке с моделью искусственной нейронной сети с помощью языка программирования Python и библиотек для работы с видеоданными и стереокамерой.

Модель управления с прогнозированием; система управления; мобильный робот; машинное зрение; нейронная сеть.

A.A. Tkachenko, D.D. Devyatkin

CONTROL SYSTEM DESIGN AND AUTONOMY FOR TWO-WHEELED MOBILE ROBOT

Model Predictive Control is an advanced process control method that used while meeting a set of constraints. From an engineering point of view, the MPC method of designing control systems is attractive, because is relatively simple in design, including for solving complex production problems. This method is similar to the classical synthesis of a control system based on a linear-quadratic controller (LQR). The key difference between MPC and LQR is that predictive control solves the optimization problem within a sliding time horizon, while the linear quadratic method used to solve the same problem over a fixed time window. The paper considers a method for constructing two-wheeled mobile robot control system using Model Predictive Control. The process of building a mathematical model of the mechanical system of the robot is given, as well as the linearization of the resulting model is performed. The basic principles of constructing a control system based on MPC for linear systems without external disturbances, as well as using an observer to assess the state of the model under the influence of additive white Gaussian noises, are presented. A variant of the synthesis of a control system with imposed restrictions on the input signal is considered. Also presented is a method for determining the position of a two-wheeled robot in space using a vision system, which is based on the use of a neural network. The architecture of the used model is given, as well as a stereo camera, which used to build an image depth map. In addition to the above, the work describes in detail the principle of the deep learning model – YOLOv3, which based on several blocks of input data processing. A detailed description of the implementation of a stereo camera in conjunction with an artificial neural network model using the Python programming language and libraries for working with video data and a stereo camera is presented.

Model Predictive Control; control system; mobile robot; computer vision; neural network.

Введение. Двухколесные мобильные роботы стали популярными проектами многих крупных компаний в наши дни. В их числе одна из самых известных, занимающаяся разработкой человекоподобных роботов – Boston Dynamics. Целью данной работы является решение задачи проектирования системы управления на основе Model Predictive Control (MPC) для линейной Multiple Input Multiple Output (MIMO) системы. Данный метод подразумевает использование в каждый момент времени информации о местоположении объекта в пространстве. Но широко известные способы позиционирования (такие как GPS, IMU или колесная одометрия) не дают достаточной точности или вовсе не пригодны для использования на мобильном роботе, который передвигается в помещении. Для решение данной проблемы, а также для осуществления возможности распознавания интересующих объектов в поле зрения робота, рассматривается вариант использования стереокамеры с технологией RealSense, которая позволяет получать 3D-изображение. Принцип действия 3D-камеры состоит в одновременном формировании двух видов изображений: обычного RGB кадра и карты глубин, получаемой с помощью инфракрасных излучателей и датчиков. Для решения проблемы детекции образов используется современная архитектура You Only Look Once (YOLO) [19, 20], которая используется для распознавания множественных объектов на изображении.

Математическая модель. Для описания динамики движения робота-балансира в пространстве необходимо вывести математическую модель. В качестве обобщенных координат возьмем координату положения центра колеса x и угол отклонения робота от вертикали φ .

На данном этапе предполагается, что робот способен совершать два типа движения: поступательное вдоль оси x и вращательное вокруг оси колеса.

Полагая, что при малых значениях φ имеют место равенства [1]

$$\sin \varphi \approx \varphi, \quad \cos \varphi \approx 1, \quad \dot{\varphi} \approx 0, \quad (1)$$

запишем систему следующим образом:

$$\left\{ \begin{array}{l} \ddot{x} = \frac{-\frac{(m_b l)^2 g \varphi}{J_b + m_b l^2} + \left(\frac{2}{r} + \frac{2m_b l}{J_b + m_b l^2} \right) M}{m_b + 2 \left(\frac{1}{r^2} J_w + m_w \right) - \frac{(m_b l)^2}{J_b + m_b l^2}} \\ \ddot{\varphi} = \frac{m_b g l \varphi - \left(\frac{2m_b l}{m_b r + 2 \left(\frac{1}{r} J_w + m_w r \right)} + 2 \right) M}{J_b + m_b l^2 - \frac{(m_b l)^2}{m_b + 2 \left(\frac{1}{r^2} J_w + m_w \right)}} \end{array} \right. \quad (2)$$

Введем вектор состояний

$$\bar{x} = [x \quad \dot{x} \quad \varphi \quad \dot{\varphi}]^T. \quad (3)$$

Теперь, используя ряд обозначений для более компактной записи, получим систему в матричной форме:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\varphi} \\ \ddot{\varphi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -C_1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & C_3 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \varphi \\ \dot{\varphi} \end{bmatrix} + \begin{bmatrix} 0 \\ C_2 \\ 0 \\ -C_4 \end{bmatrix} M. \quad (4)$$

Далее полученную модель можно уточнить с учетом использованной силовой установки, а именно двигателей постоянного тока с редуктором [2]. Учитывая специфику и конструктивные постоянные двигателей, запишем уточненную математическую модель [3–6]:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\varphi} \\ \ddot{\varphi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -C_2 \alpha & -C_1 & C_2 r \alpha \\ 0 & 0 & 0 & 1 \\ 0 & C_4 \alpha & C_3 & -C_4 r \alpha \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \varphi \\ \dot{\varphi} \end{bmatrix} + \begin{bmatrix} 0 \\ C_2 \beta \\ 0 \\ -C_4 \beta \end{bmatrix} U. \quad (5)$$

Здесь α и β выражаются следующим образом

$$\alpha = \frac{N_G k_m k_e}{rR}, \quad \beta = \frac{N_G k_m}{R}, \quad (6)$$

где N_G – передаточное отношение, R – сопротивление обмоток ротора, k_e , k_m – конструктивные постоянные двигателя, r – радиус.

Далее будет приводиться описание построения системы управления на основе МРС непосредственно для выведенной ранее математической модели робота с соответствующими матрицами. При использовании модели пространства состояний текущая информация, необходимая для прогнозирования, представлена переменной состояния в текущий момент времени.

Формирование расширенной модели. Принцип построения системы управления на основе МРС базируется на формировании новой расширенной модели исходной системы.

Преобразуя исходную систему к системе дискретного времени, получим общий вид расширенной модели будет иметь вид

$$\begin{bmatrix} \Delta x_m[k+1] \\ y[k+1] \end{bmatrix} = \begin{bmatrix} A_m & o_m^T \\ C_m A_m & I_{q \times q} \end{bmatrix} \begin{bmatrix} \Delta x_m[k] \\ y[k] \end{bmatrix} + \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \Delta u[k], \quad (7)$$

$$y[k] = \begin{bmatrix} o_m & I_{q \times q} \end{bmatrix} \begin{bmatrix} \Delta x_m[k] \\ y[k] \end{bmatrix},$$

где $I_{q \times q}$ – единичная матрица размерности $q \times q$, а o_m – нулевая матрица размерности $q \times n$.

Перепишем систему в более удобном и привычном виде:

$$\begin{aligned} x[k+1] &= Ax[k] + B\Delta u[k], \\ y[k] &= Cx[k], \end{aligned} \quad (8)$$

где A, B, C – матрицы, соответствующие новой расширенной системе.

Принимая во внимание то, что в нашем случае матрицы дискретного аналога системы (5) таковы, что

$$A_m = \begin{bmatrix} 1 & 0,0212 & 0,0007 & 0,0031 \\ 0 & 0,1695 & 0,0637 & 0,0339 \\ 0 & 0,4193 & 1,1833 & 0,0896 \\ 0 & 4,8881 & 3,5254 & 0,9878 \end{bmatrix}, B_m = \begin{bmatrix} 0,0183 \\ 0,1928 \\ -0,0973 \\ -1,1348 \end{bmatrix}, \quad (9)$$

$$C_m = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

то искомые матрицы A, B, C соответствующие расширенной модели, имеют вид

$$A = \begin{bmatrix} 1 & 0,0212 & 0,0007 & 0,0031 & 0 & 0 \\ 0 & 0,1695 & 0,0637 & 0,0339 & 0 & 0 \\ 0 & 0,4193 & 1,1833 & 0,0896 & 0 & 0 \\ 0 & 4,8881 & 3,5254 & 0,9878 & 0 & 0 \\ 1 & 0,0212 & 0,0007 & 0,0031 & 1 & 0 \\ 0 & 0,4193 & 1,1833 & 0,0896 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0,0183 \\ 0,1928 \\ -0,0973 \\ -1,1348 \\ 0,0183 \\ -0,0973 \end{bmatrix}, \quad (10)$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Построение прогнозирующего управления. После формулировки расширенной модели, следующим шагом в проектировании системы прогнозирующего управления является расчет будущего выхода системы. Этот процесс описывается в окне оптимизации [7, 8]. Далее будет подробно рассмотрена оптимизация, выполняемая в этом окне. Будем полагать, что текущий момент k_i , а N_p – длина оптимизационного окна (горизонт прогнозирования), определяющая количество будущих состояний.

Пусть вектор состояний $x[k_i]$ полностью известен из измерений. Прогнозируемую траекторию управления можно представить следующим образом

$$\Delta u[k_i], \Delta u[k_i + 1], \dots, \Delta u[k_i + N_c - 1], \quad (11)$$

где N_c – горизонт управления, определяющий количество параметров, используемых для построения траектории управления. Важно отметить, что $N_c \leq N_p$.

На основе расширенной модели выражения для вычисления набора будущих состояний системы имеют вид

$$\begin{aligned} x[k_i + 1 | k_i] &= Ax[k_i] + B\Delta u[k_i] \\ x[k_i + 2 | k_i] &= A^2x[k_i] + AB\Delta u[k_i] + B\Delta u[k_i + 1] \\ &\vdots \\ x[k_i + N_p | k_i] &= A^{N_p}x[k_i] + A^{N_p-1}B\Delta u[k_i] + \\ &\quad + \dots + A^{N_p-N_c}B\Delta u[k_i + N_c - 1]. \end{aligned} \quad (12)$$

С помощью набора (9) можно спрогнозировать выходы для системы:

$$\begin{aligned} y[k_i + 1 | k_i] &= CAx[k_i] + CB\Delta u[k_i] \\ y[k_i + 2 | k_i] &= CA^2x[k_i] + CAB\Delta u[k_i] + CB\Delta u[k_i + 1] \\ &\vdots \\ y[k_i + N_p | k_i] &= CA^{N_p}x[k_i] + CA^{N_p-1}B\Delta u[k_i] + \\ &\quad + \dots + CA^{N_p-N_c}B\Delta u[k_i + N_c - 1]. \end{aligned} \quad (13)$$

Определим векторы

$$Y = [y[k_i + 1 | k_i] \quad y[k_i + 2 | k_i] \quad \dots \quad y[k_i + N_p | k_i]]^T \quad (14)$$

и

$$\Delta U = [\Delta u[k_i]^T \quad \Delta u[k_i + 1]^T \quad \dots \quad \Delta u[k_i + N_c - 1]^T]^T, \quad (15)$$

где N_p и N_c – размерности Y и ΔU соответственно.

Таким образом, имеем

$$Y = Fx[k_i] + \Phi\Delta U, \quad (16)$$

где

$$F = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N_p} \end{bmatrix}, \Phi = \begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & \dots & CA^{N_p-N_c}B \end{bmatrix}. \quad (17)$$

Оптимальный вектор управления в оптимизационном окне задается следующим образом [7]:

$$\Delta U = (\Phi^T\Phi + R)^{-1}(\Phi^TR_s r[k_i] - \Phi^TFx[k_i]), \quad (18)$$

где R – диагональная матрица, содержащая весовой параметр r_w и определяемая как $R = r_w I$, где матрица I , как и $\Phi^T\Phi$ имеет размерность $mN_c \times mN_c$, Φ^TF имеет размерность $mN_c \times n$, а Φ^TR_s соответствует последним q столбцам Φ^TF . Матрица R_s определяется как $R_s = [I_{q \times q} \quad I_{q \times q} \quad \dots \quad I_{q \times q}]^T$ и имеет размер $1 \times N_p$. Желаемый сигнал уставки имеет вид: $r[k_i] = [r_1[k_i] \quad r_2[k_i] \quad \dots \quad r_q[k_i]]^T$.

Так как принцип управления со скользящим горизонтом подразумевает использование только первого элемента вектора ΔU в момент времени k_i в качестве управляющего воздействия, то получим

$$\begin{aligned} \Delta u[k_i] = & [I_m \quad o_m \quad \dots \quad o_m](\Phi^T \Phi + R)^{-1} \times \\ & \times (\Phi^T R_s r[k_i] - \Phi^T Fx[k_i]) = K_y r[k_i] - K_{mpc} x[k_i]. \end{aligned} \quad (19)$$

Ограничения как часть оптимального решения. Поскольку необходимо наложить ограничения на входящее воздействие, которым является источник питания, необходимо выразить ограничения в виде системы неравенств на основе вектора параметров ΔU , а затем объединить их с исходной функцией стоимости J . Рассматривая нижний и верхний пределы, запишем неравенства в матричной форме:

$$\begin{bmatrix} -I \\ I \end{bmatrix} \Delta U \leq \begin{bmatrix} -\Delta U^{\min} \\ \Delta U^{\max} \end{bmatrix}. \quad (20)$$

Ограничения накладываются на все будущие элементы выборки, поэтому для регулируемого параметра будем иметь следующие неравенства

$$\begin{aligned} -(C_1 u(k_i - 1) + C_2 \Delta U) & \leq -U^{\min}, \\ (C_1 u(k_i - 1) + C_2 \Delta U) & \leq U^{\max}, \end{aligned} \quad (21)$$

где U^{\min} и U^{\max} являются столбцами, содержащими N_c элементов u^{\min} и u^{\max} соответственно.

Ограничения на выходной сигнал, в таком случае, примут вид

$$Y^{\min} \leq Fx(k_i) + \Phi \Delta U \leq Y^{\max}. \quad (22)$$

Модель прогнозирующего управления при наличии ограничений будет строиться путем нахождения вектора параметров ΔU , который минимизирует следующий функционал [9]

$$\begin{aligned} J = & (R_s - Fx(k_i))^T (R_s - Fx(k_i)) - \\ & - 2\Delta U^T \Phi^T (R_s - Fx(k_i)) + \Delta U^T (\Phi^T \Phi + R) \Delta U, \end{aligned} \quad (23)$$

с учетом ограничений. Поскольку данный функционал является квадратичным, а ограничения представляют собой линейные неравенства, проблема нахождения оптимального управления становится проблемой нахождения оптимального решения стандартной задачи квадратичного программирования.

Моделирование системы управления. Для проектирования системы управления воспользуемся полученными ранее выкладками. Положим $N_c = 10$, $N_p = 20$, значение уставки на всем временном отрезке равно $r = [1 \ 0]^T$, т.е. наша цель добиться нулевого отклонения тела робота от вертикали и сместить его на 1 м. Вектор состояний исходной системы, а также вектор состояний расширенной модели примем в начальный момент нулевыми. Определим оптимальное решение ΔU при $r_w = 0,5$.

Из приведенных на рис. 1. графиков следует, что выходной сигнал u достигает и принимает значение заданной величины параметра уставки.

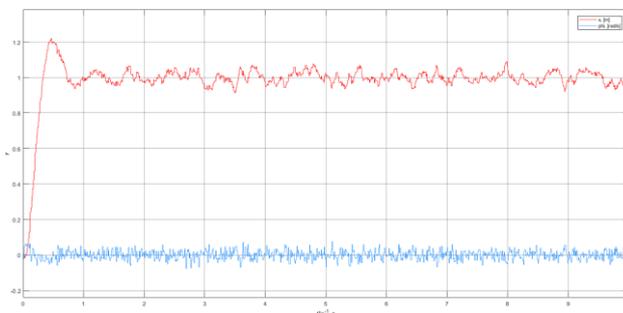


Рис. 1. Изменение выходного сигнала

Распознавание образов. На сегодняшний день существует несколько архитектур искусственных нейронных сетей, созданных для распознавания объектов [10, 11]. Эти архитектуры в основном делятся на «двухуровневые» и «одноуровневые». «Двухуровневые» нейронные сети, такие как RCNN, fast-RCNN и faster-RCNN, используют регионы на изображении, чтобы определить присутствие определенного объекта в нем. В таких моделях есть две ключевые проблемы: они не смотрят на изображение «полностью», а только на отдельные регионы, и скорость обработки изображений мала, так как обрабатывает одно изображение несколько тысяч раз. «Одноуровневые» модели, такие как YOLO, SSD и FPN, имеют другой принцип работы по сравнению с предыдущими, они не используют регионы вообще, обрабатывает изображение целиком и очень быстрые, потенциально пригодные для работы в реальном времени. Для детекции объектов используется архитектура YOLOv3. Главная особенность этой архитектуры состоит в том, что операция дискретной свертки часть архитектуры применяется один раз ко всему изображению сразу [12]. Нейронная сеть разделяет изображение на сетку и предсказывает ограничивающие рамки и вероятности того, что там присутствует некоторый объект для каждого участка.

В качестве входного сигнала в модели используется RGB изображение - признаковое описание входа, на выходе нейронной сети получается метка класса, которая соответствует объекту на изображении, если он там присутствует, и координаты обрамляющих рамок этого объекта.

Перед обработкой изображения нейронной сетью оно приводится к размеру 416×416 . Далее изображение разбивается на $S \times S$ ячеек. Изображение делится на 3 разных сетки: 8×8 , 16×16 и 32×32 , для лучшего распознавания объектов разного размера, т.е. обнаружение объектов выполняется на 3-х масштабах. Для извлечения признаков используется уникальная архитектура сверточной топологии, которая состоит из 53 слоев, с названием Darknet53. Данная архитектура содержит сверточные слои с фильтрами 3×3 и 1×1 , в качестве активации LeakyReLU, а также присутствуют слои BN и residual [13, 14]. Эта подсеть уменьшает разрешение изображения в 8 раз, 16 раз и 32 раза. Следующий шаг – агрегация информации от отдельных слоев с предыдущих блоков с помощью Path Aggregation Network или др.

Далее модель прогнозирует ограничивающие рамки, используя кластеры измерений в качестве якорных рамок. Поля якорных рамок выбираются на основе размеров объекта в обучающих наборах данных, для подбора их параметров использовался алгоритм k-means и метрика IoU. После этого для каждой ячейки модель получает:

1. Позицию каждой якорной рамки относительно ячейки.
2. Вероятностное распределение для каждого класса.
3. Оценку уверенности в якорной рамке по метрике IoU.

$$\begin{aligned}
 b_x &= \sigma(t_x) + c_x, \\
 b_y &= \sigma(t_y) + c_y, \\
 b_w &= p_w e^{t_w}, \\
 b_h &= p_h e^{t_h}, \\
 \sigma(t_0) &= P(obj) \cdot IoU(b, obj),
 \end{aligned}
 \tag{24}$$

где (t_x, t_y) – координаты левого верхнего угла, (t_w, t_h) – ширина и высота, спрогнозированные моделью, c_x, c_y – координаты верхней левой точки ячейки, с которой происходит работа, p_w, p_h – ширина и высота якорной рамки, b_x, b_y, b_w, b_h – значения, определяющие ограничивающую рамку (смещение), $\sigma(t_0)$ – качество якорной рамки, $P(obj)$ – вероятность конкретного объекта в рамке. На рис. 2 изображен пример ограничивающей рамки.

Далее происходит фильтрация рамок по вероятности нахождения в них объектов с помощью алгоритма подавления ложных максимумов, который заключается в следующем:

- 1) рассматривается ограничивающая рамка с наибольшей вероятностью принадлежности к объекту;
- 2) рассматриваются все ограничивающие рамки, которые тоже относятся к этому объекту;
- 3) если метрика IoU с первой ограничивающей рамкой больше заданного порога, то рамка удаляется.

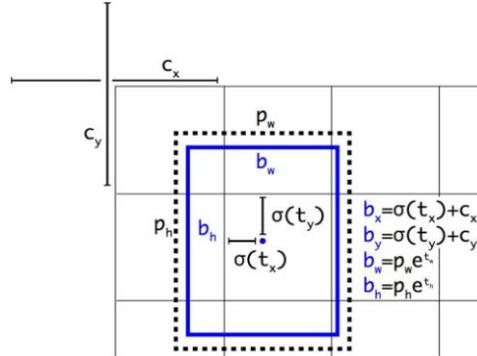


Рис. 2. Визуализация ограничивающей рамки

В архитектуре YOLOv3 классическая функция активации softmax заменяется на независимые логистические классификаторы [15], т.к. выходные метки не являются взаимоисключающими. Потери классификации вычисляются на основе бинарной кросс-энтропии.

Функция потерь в архитектуре YOLOv3 состоит из:

- 1) ошибок классификации:

$$-\sum_{i=0}^{S^2} \mathbf{1}_i^{obj} \sum_{c \in \text{classes}} y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i),
 \tag{25}$$

где S^2 – количество ячеек в сетке, $\mathbf{1}_i^{obj} = 1$, если объект obj появляется в ячейке i , иначе 0, \hat{y}_i – спрогнозированная метка класса в ячейке i , y_i – истинная метка класса;

2) ошибок локализации объекта на изображении:

$$\lambda \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] + \lambda \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right], \quad (26)$$

где $1_{ij}^{obj} = 1$, если ограничивающая рамка j в ячейке i отвечает за обнаружение объекта, иначе 0, B – количество рамок, (x_i, y_i) и \hat{x}_i, \hat{y}_i – координаты левого верхнего угла для истинной и предсказанной рамки, (w_i, h_i) и \hat{w}_i, \hat{h}_i – ширина и высота истинной и предсказанной рамки;

3) ошибок уверенности в объекте:

$$\sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i)^2, \quad (27)$$

если объект обнаружен, и

$$\alpha \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{noobj} (C_i - \hat{C}_i)^2, \quad (28)$$

если объект не обнаружен. Здесь \hat{C}_i – показатель достоверности рамки j в ячейке i , $1_{ij}^{obj} = 1$, – если j -й ограничивающий прямоугольник в ячейке i отвечает за обнаружение объекта, иначе 0, $1_{ij}^{noobj} = \bar{1}_{ij}^{obj}$, α – коэффициент уменьшения ошибки.

В силу того, что данная архитектура способна обрабатывать кадры в режиме реального времени, это дает огромный потенциал, в совокупности с камерой Intel RealSense D435i, для навигации мобильных роботов в относительно небольших пространствах, таких как офисы, квартиры и др. Использование этих инструментов позволяет распознавать необходимый объект и вычислять дистанцию до него, чтобы корректно подавать сигнал на регулятор управления.

Реализация. Для моделирования использовался язык программирования Python [16] и 2 библиотеки: `opencv` и `pyrealsense2` [17, 18]. Первая предназначена для обработки видеоданных, вторая позволяет использовать функционал камеры. При использовании этих библиотек получается видеопотока с камеры, который представляется в виде 2-х матриц: матрица RGB изображения и матрица расстояний до каждого пикселя изображения. Далее происходит детекции объекта на матрице RGB изображения, вычисляется центр ограничивающей рамки и до этой точки считается расстояние.

Заключение. В данной работе была получена математическая модель, описывающая динамику движения двухколесного робота. Затем был реализован синтез оптимального управления для стабилизации робота на основе Model Predictive Control (MPC) при влиянии аддитивных белых гауссовских шумов, а также с учетом накладываемых ограничений на входное воздействие. Реализация архитектуры YOLOv3 в синтезе с камерой Intel RealSense D435i.

Важной особенностью данной работы, подчеркивающей ее актуальность, является применение рассматриваемого подхода к проектированию автономного робота на физической модели, которая была сконструирована и в данный момент отлаживается.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Дронг В.И., Дубинин В.В., Ильин М.М. и др. Курс теоретической механики: Учебник для вузов / под общ. ред. Колесникова К.С. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2005. – 736 с.
2. Иродов И.Е. Электromagnetизм. Основные законы. – М.: БИНОМ. Лаборатория знаний, 2014. – 319 с.
3. Андриевский Б.Р., Фрадков А.Л. Избранные главы теории автоматического управления с примерами на языке Matlab. – СПб.: Наука, 2000. – 475 с.
4. Воронов А.А. Устойчивость, управляемость, наблюдаемость. – М.: Главная редакция физико-математической литературы: Изд-во «Наука», 1979. – 336 с.
5. Заде Л., Дезоер Ч. Теория линейных систем. Метод пространства состояний. – М.: Наука, 1970. – 704 с.
6. Ройтенберг Я.Н. Автоматическое управление. – М.: Наука, 1971. – 396 с.
7. Liuping Wang. Model Predictive Control System Design and Implementation Using MATLAB. – Springer-Verlag London Limited, 2009. – 403 p.
8. Ridong zhang, Anke Xue, Furong Gao. Model Predictive Control. Approaches Based on the Extended State Space Model and Extended Non-minimal State Space Model. – Springer Nature Singapore Pte Ltd, 2019. – 143 p.
9. James B. Rawlings, David Q. Mayne, Moritz M. Diehl. Model Predictive Control: Theory, Computation, and Design. – 2nd ed. Copyright 2017 by Nob Hill Publishing, LLC. Nob Hill Publishing, LLC Cheryl M. Rawlings, publisher Madison, WI 53705.
10. Девяткин Д.Д., Порцев Р.Ю., Макаренко А.В. Сравнение 3D-сверточных нейронных сетей с полносвязными и GAP слоями в задаче распознавания действий на видео // Тр. 17-й Всероссийской школы-конференции молодых ученых «Управление большими системами» (УБС'2021, Москва). – М.: ИПУ РАН, 2021. – С. 190-201.
11. Лепский А.Е., Броневич А.Г. Математические методы распознавания образов: Курс лекций. – Таганрог: Изд-во ТТИ ЮФУ, 2009. – 155 с.
12. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение: пер. с англ. А.А. Слинкина. – 2-е изд., испр. – М.: ДМК Пресс, 2018. – 652 с.
13. Szegedy Christian, Liu Wei, Jia Yangqing, Sermanet Pierre, Reed Scott, Anguelov Dragomir, Erhan Dumitru, Vanhoucke Vincent, Rabinovich Andrew Going deeper with convolutions. ArXiv:1409.4842, 2014.
14. Shtonyan Karen, Zisserman Andrew Very deep convolutional networks for large-scale image recognition. ArXiv: 1409.1556, 2014.
15. Рашка С. Python и машинное обучение: пер. с англ. А.В. Голунова. – М.: ДМК Пресс, 2017. – 418 с.
16. Любанович Б. Простой Python. Современный стиль программирования. – СПб.: Питер, 2016. – 480 с.
17. Жерон Орельен. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем: пер. с англ.
18. Гадушкин А. Синтез многослойных систем распознавания образов. – М.: Энергия, 1974. – 368 с.
19. Joseph Redmon, Ali Farhadi. Yolo9000: Better, faster, stronger. In Computer Vision and Pattern Recognition (CVPR) // 2017 IEEE Conference on. – IEEE, 2017. – P. 6517-6525.
20. Redmon Joseph, Farhadi Ali. Yolov3: An incremental improvement, 2018.

REFERENCES

1. Drong V.I., Dubinin V.V., Il'in M.M. i dr. Kurs teoreticheskoy mekhaniki: Uchebnik dlya vuzov [Course of theoretical mechanics: textbook for universities], under the general ed. Kolesnikova K.S. Moscow: Izd-vo MGTU im. N.E. Bauman, 2005, 736 p.
2. Irodov I.E. Elektromagnetizm. Osnovnye zakony [Electromagnetism. Basic laws]. Moscow: BINOM. Laboratoriya znaniy, 2014, 319 p.
3. Andrievskiy B.R., Fradkov A.L. Izbrannye glavy teorii avtomaticheskogo upravleniya s primerami na yazyke Matlab [Selected chapters of the theory of automatic control with examples in Matlab]. Saint Petersburg: Nauka, 2000, 475 p.
4. Voronov A.A. Ustoychivost', upravlyaemost', nablyudaemost' [Stability, controllability, observability]. Moscow: Glavnaya redaktsiya fiziko-matematicheskoy literatury: Izd-vo «Nauka», 1979, 336 p.

5. *Zade L., Dezoer Ch.* Teoriya lineynykh sistem. Metod prostranstva sostoyaniy [Theory of linear systems. The method of the space of states]. Moscow: Nauka, 1970, 704 p.
6. *Roytenberg Ya.N.* Avtomaticheskoe upravlenie [Automatic control]. Moscow: Nauka, 1971, 396 p.
7. *Liuping Wang.* Model Predictive Control System Design and Implementation Using MATLAB. Springer-Verlag London Limited, 2009, 403 p.
8. *Ridong zhang, Anke Xue, Furong Gao.* Model Predictive Control. Approaches Based on the Extended State Space Model and Extended Non-minimal State Space Model. Springer Nature Singapore Pte Ltd, 2019, 143 p.
9. *James B. Rawlings, David Q. Mayne, Moritz M. Diehl.* Model Predictive Control: Theory, Computation, and Design. 2nd ed. Copyright 2017 by Nob Hill Publishing, LLC. Nob Hill Publishing, LLC Cheryl M. Rawlings, publisher Madison, WI 53705.
10. *Devyatkin D.D., Portsev R.Yu., Makarenko A.V.* Sravnenie 3D-svertochnykh neyronnykh setey s polnosvyaznymi i GAP sloyami v zadache raspoznavaniya deystviy na video [Comparison of 3D convolutional neural networks with fully connected and GAP layers in the task of recognizing actions on video], *Tr. 17-y Vserossiyskoy shkoly-konferentsii molodykh uchenykh «Upravlenie bol'shimi sistemami» (UBS'2021, Moskva)* [Proceedings of the 17th All-Russian School-Conference of Young Scientists "Management of Large Systems" (UBS'2021, Moscow)]. Moscow: IPU RAN, 2021, pp. 190-201.
11. *Lepskiy A.E., Bronevich A.G.* Matematicheskie metody raspoznavaniya obrazov: Kurs lektsiy [Mathematical methods of pattern recognition: A course of lectures]. Taganrog: Izd-vo TTI YuFU, 2009, 155 p.
12. *Gudfellow Ya., Bendzhio I., Kurvill' A.* Glubokoe obuchenie [Deep learning]: transl. from the engl. by A.A. Slinkina. 2 ed. Moscow: DMK Press, 2018, 652 p.
13. *Szegedy Christian, Liu Wei, Jia Yangqing, Sermanet Pierre, Reed Scott, Anguelov Dragomir, Erhan Dumitru, Vanhoucke Vincent, Rabinovich Andrew* Going deeper with convolutions. ArXiv:1409.4842, 2014.
14. *Simonyan Karen, Zisserman Andrew* Very deep convolutional networks for large-scale image recognition. ArXiv: 1409.1556, 2014.
15. *Rashka S.* Python i mashinnoe obuchenie [Python and machine learning]: transl. from the engl. by A.V. Golunova. Moscow: DMK Press, 2017, 418 p.
16. *Lyubanovich B.* Prostoy Python. Sovremennyy stil' programmirovaniya [Simple Python. Modern programming style]. Saint Petersburg: Piter, 2016, 480 p.
17. *Zheron Orel'en.* Prikladnoe mashinnoe obuchenie s pomoshch'yu Scikit-Learn i TensorFlow: konseptsii, instrumenty i tekhniki dlya sozdaniya intellektual'nykh system [Applied machine learning using Scikit-Learn and TensorFlow: concepts, tools and techniques for creating intelligent systems]: transl. from the engl.
18. *Galushkin A.* Sintez mnogosloynnykh sistem raspoznavaniya obrazov [Synthesis of multilayer image recognition systems]. Moscow: Energiya, 1974, 368 p.
19. *Joseph Redmon, Ali Farhadi.* Yolo9000: Better, faster, stronger. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. IEEE, 2017, pp. 6517-6525.
20. *Redmon Joseph, Farhadi Ali.* YoloV3: An incremental improvement, 2018.

Статью рекомендовал к опубликованию к.ф.-м.н. А.Ю. Кустов.

Ткаченко Андрей Александрович – Институт проблем управления им. В.А. Трапезникова; e-mail: tkachenko_aa10@mail.ru; г. Москва, Россия; тел.: +79255817122; математик.

Девяткин Даниил Дмитриевич – e-mail: danya.devyatkin@mail.ru; тел.: +79035883027; математик.

Tkachenko Andrew Alexandrovich – Institute of Control Sciences V.A. Trapeznikov RAS; e-mail: tkachenko_aa10@mail.ru; Moscow, Russia; phone: +79255817122; mathematician.

Devyatkin Daniil Dmitrievich – e-mail: danya.devyatkin@mail.ru; phone: +79035883027; mathematician.