Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

A Hovercraft Testbed for Decentralized and Cooperative Control

Vladimeros Vladimerou† , Andrew Stubbs, Joel Rubel, Adam Fulford, Jeffrey Strick, Geir Dullerud
Coordinated Science Laboratory, University of Illinois, Urbana, IL 61801

***Abstract*— This paper describes a testbed facility – the HoTDeC (HOvercraft Testbed for DEcentralized Control) – developed by the authors at the University of Illinois, consisting of multiple autonomous hovercraft vehicles which are wirelessly networked. This facility provides a flexible and state-of-the-art testbed for experimentation with inter-networked vehicles and sensors for decentralized and cooperative control, in a dynamically nontrivial setting.**

I. INTRODUCTION

The HoTDeC consists of several floating autonomous vehicles, hovercraft, which receive visual feedback from a system of networked overhead cameras. A wireless network configuration allows the hovercraft to communicate with each-other and with a desktop computer, connected to the overhead camera system. The vehicles themselves have on-board computers that are responsible for communications and actuation of their thrusters.

The project design is continuously evolving. It has originated from a testbed, designed by the same group, that ran on a modified air-hockey table as presented in [4], [5] and [6]. Current information can be obtained from our website[1].

Several other similar networked testbeds exist and they are referenced in [9], [10] and [11].

Fig. 1. Hovercraft fleet in formation

However, the usefulness and uniqueness of our testbed is based on a number of characteristics:

The testbed allows for *Internet-based vehicle software delivery* and *real-time task configuration*. The operating system software on the vehicles can be updated from anywhere on the web, through a wireless Ethernet connection and controllers can potentially be loaded. Since Internet users can monitor their test-runs via a web-cam, controller parameters can be modified on-line, in real-time. This makes HoTDeC a remotely operated control testbed. There are safety measures taken by the testbed administrators on top of the user-run algorithms.

The vehicles are *open-loop marginally stable, fully actuated.* The nature of our hovercraft allows for simulation of models with features found in many air vehicles. Hovercraft, unlike wheeled vehicles, have much less open-loop stability. Due to the nature of the hovercraft fans, peer drag/thrust effects can also be tested in a multi-vehicle environment where co-operation and formation is required.

The system is decentralized, and a number of its components are wirelessly connected. The wireless connections enable all craft and networked computers to be interconnected. Controllers run on-board, so the system is decentralized. The vehicles only get vision feedback during the control process, but can also be configured through the same network (when docked, or when the configuration-required bandwidth is small enough)

The *hovercraft function autonomously* with the help of a *full-featured computer on-bard*. The vehicle has the full capabilities of a desktop computer case on-board. Peripheral cards can be added, enabling the use of on-board sensors or extra computing power. The computer system is stand-alone and can drive a number of conventional electronic actuators and sensors on any type of vehicle.

Our system runs *customized hardware.* A dual processor system (including a x86 compatible processor and an MCU) has access to numerous I/O ports. An MCU-interfacing board made in-house connects to the main SBC (single board computer). The whole system provides a plethora of I/O including PWM outputs, TTL logic, 3.3V logic, A/D converters, ISA bus, PCI bus, USB, Ethernet etc. Hard-real-time processes run on 2 on-board processors: the main Transmetta Crusoe CPU and the MCU board's processor. The on-board software allows for compiled controller modules to be uploaded and monitoring software that ensures safety of the vehicles and environment against user controllers.

A *system recharging module* allows *extended uptime*, and uses *autonomously mounted recharging stations*. This recently added feature enables the system to run continuously unattended. The vehicles automatically dock to recharging stations when they run out of battery power. This reduces the percentage of time during which there are no hovercraft running, and allows the system to be free of regular human attendance

The *totally modular* nature of the system allows to

† Corresponding Author: `vladimer@uiuc.edu`
[1] `http://legend.me.uiuc.edu/hotdec`

expand any aspect of it, interchange modules with other systems. A different vehicle chassis/mechatronics configuration allows the system to run outdoors on a reasonably smooth floor. Even more, the electronics can be attached to most types of vehicles with mechatronic systems and function adequately with some software modifications.

II. NETWORKING AND COMMUNICATIONS

The HoTDeC system network consists of wired and wireless nodes and is connected to the Internet. All vehicles have both bluetooth (via USB) and wi-fi (via PCMCIA) interfaces. The on-board computers also have regular Ethernet ports which were only needed for initial development. The diagram shown in fig. 5 shows how the system nodes are connected.

The hovercraft use either IEEE 802.11 (wi-fi) or Bluetooth, at a given time, to communicate with the gateways. Both Bluetooth and wi-fi can be used for control, but only wi-fi is used for transferring long data streams to the hovercraft and only when all bluetooth modules are off. Wi-fi signal quality deteriorates when there are Bluetooth devices running in its range. Both protocols' delays are below our vision sampling latencies.

Our wi-fi network is private and can operate seamlessly independent of the MIE[2] department's wireless system. We can also use the MIE wireless network for our hovercraft. The lab, vehicle, and sensor gateways can all be implemented on a single desktop computer in our lab. We currently use only one machine.

Whereas the *vehicle gateway* software conveys feedback data from the sensors to the controllers on the hovercraft, the *lab gateway* offers Internet access and allows for remote reconfigurability and some on-line parameter control. These capabilities are offered via a web-server and Java applets and servlets.

Camera data is combined on the *sensor gateway* and presented to the *vehicle gateway* for transmission to the hovercraft.

III. SYSTEM CONFIGURATION OVER NETWORK

A main feature of HoTDeC is its remote *user*[3] connectivity and internetwork-based reconfigurability. A single *user* (the creators of the testbed) currently is capable for downloading, verifying and running controllers on the hovercraft. With database management of potential users, more than one remote researchers can time-share control of the testbed. At each given time, only one user can implement controllers - the vehicle cluster can only be run as a whole.

A. Network Use

To allow for safety during use, pre-emptive controllers must be run when the system is in a critical state. Since checking for safety has not yet implemented in the lower level, the only *user* defined commands available are high-level ones. For example, a *user* can tell the hovercraft to stabilize at a given point, or follow a trajectory, but the lower-level controller (that actually produces the right thrust values for the fans) cannot be changed and it is maintained by the testbed administrators. Therefore, a *user* "controller" is not in the same context as a classical controller. It is a path planning configuration that uses a built-in classical controller

[2]Mechanical and Industrial Engineering

[3]the party assigned to use the testbed

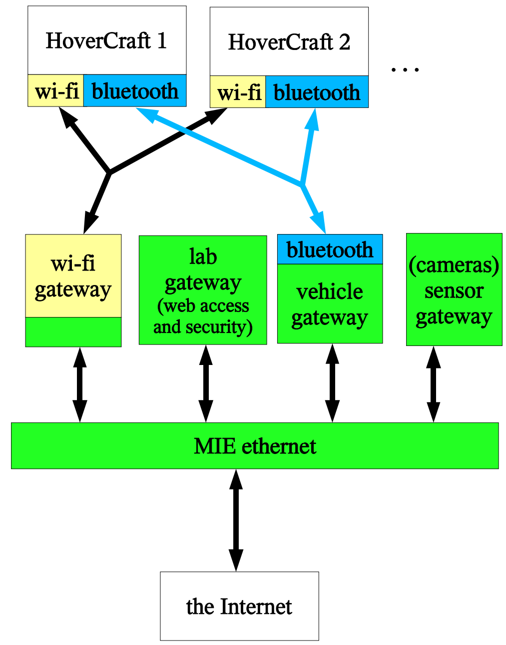

Fig. 2. Network Layer Big Picture

B. Network Configuration

Since vision data is available by default, the only configuration the *user* can enforce regarding feedback, is to choose when and where to send the vision data. Both wi-fi and bluetooth support broadcast, if it is required. A *user* can set any parameters in their "controller" to be configurable and change them in real-time. A web applet, for example, allows the *user* to mouse-click on a point located on a 2D system map, shown on their web-applet, to command a craft to go to the given point, or to command a group of craft to perform a formation regarding that point. Other similar real-time commands are available.

IV. NETWORKED SENSORS

A. Camera Specifications and System

The camera system is used to determine the position and orientation of each hovercraft while in operation. This information is then broadcast wirelessly to the hovercraft to allow each craft to calculate the appropriate control actions.

The current version of the vision system consists of an array of IEEE 1394 (FireWire) web cameras. The IEEE 1394 web cameras currently in use are iBOT FireWire Web Cams by Orange Micro. This web camera model is able to provide a 640 by 480 pixel image at 30 frames per second using a quarter-inch color CCD image sensor. The cameras

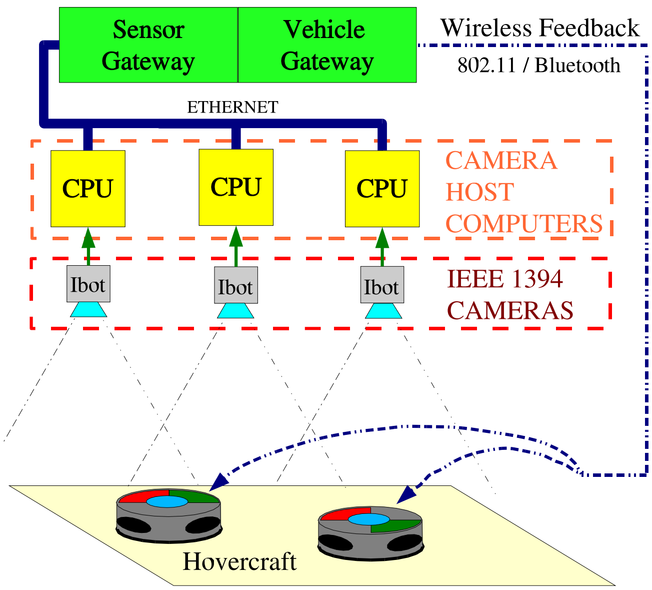

Fig. 3. Vision System

are arranged on the ceiling and aligned in a grid such that the images from each camera overlap at the edges. This creates a large operating area for the hovercraft and allows them to seamlessly move from the viewable area of one camera to another.

Each camera in the array is connected to and operated by a Camera Host Computer. The Camera Host Computers are Linux computers running software that processes the incoming images and communicates with the *Sensor Gateway*. The Camera Host Computers connected to each IEEE 1394 camera send the position and orientation of each hovercraft in viewable area to the *Sensor Gateway* via the Ethernet using the User Datagram Protocol (UDP). The *Sensor Gateway* then combines the information from the various Camera Host Computers and through the *vehicle gateway* sends the position and orientation information wirelessly to the hovercraft on a wireless connection.

B. Vision Algorithm

The images from each camera are captured and processed in real time on the Camera Host Computers using a C code program. There are several steps in finding and locating the hovercraft in each image.

First, given the distortion caused by the lens of the web camera, a set of sample images of a black and white checkerboard were used to determine the parameters required to correct for the radial distortion. The parameters were found by using the Camera Calibration Toolbox for Matlab [12]. The distorted nonlinear image was then mapped to real world coordinates.

The second step in finding the hovercraft is training the vision system to recognize the various indicating colors used on the hovercraft. This process was done by taking a set of sample colors and scanning certain portions of the image to determine the numerical range for each color given the current camera settings and environment conditions.

After calibrating the colors and adjusting for distortion, the hovercraft is easily located while in operation by scanning a grid of properly spaced image points for the large circle on the hovercraft top. After finding a point within the circle on the hovercraft top, the edges of the circle are found in the image by iterating left, right, up, and down. This process provides four points on the circumference of the circle. The four edge points are used to geometrically determine the center of the craft by creating two chords. The result is that the perpendicular bisectors of the two chords intersect at the center of the circle which indicates the craft center. Now knowing where the craft is located, the orientation and identity of the craft are determined by checking the colors and positions of the four colored quadrants.

V. REALTIME ONBOARD SYSTEM

A. Hardware and I/O Specification

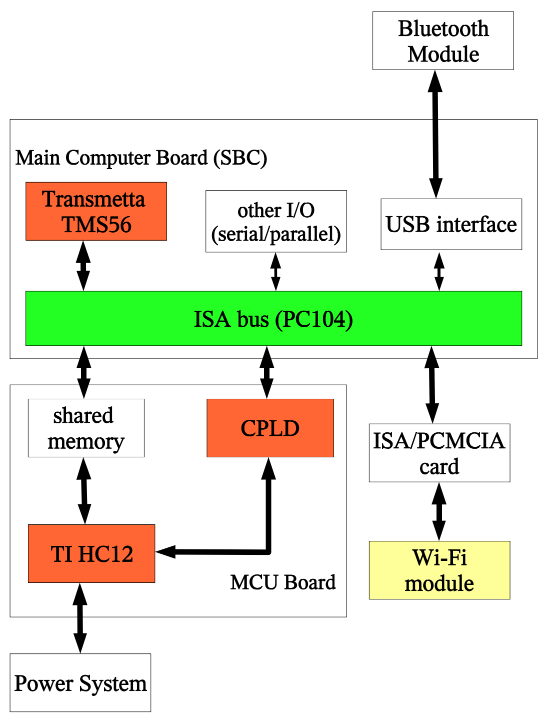

Fig. 4. Computer Systems

1) Main Computer: Shown as part of fig. 4, the single board computer on each hovercraft (Main Computer Board - SBC), is a full-function x86-compatible machine that has all features of a desktop motherboard. This includes all the default I/O, as well as expansion card slots for ISA and PCI cards (using its PC104 and PC104+ form factor buses respectively). It uses its USB interface to access a bluetooth module and an ISA/PCMCIA expansion card to access an 802.11b card. It is connected to an MCU board via its ISA bus, through shared memory.

2) MCU board: The MCU board, designed by students of our group, features a micro-controller unit and a complex programmable-logic device (CPLD) which provide the overall computer system with a number of various I/O ports, in addition to the standard PC I/O offerd on the Main Computer. It connects through PWM outputs to MOSFET switch/break units that control each thruster. It receives input from hall effect sensors on the current RPM reading of the fan. It is also connected to the system's power board, allowing it to read and convey battery voltage and current readings to the Main Computer.

B. Software

The Main Computer handles communications and the main control loop, as well as makes decisions regarding charging, watchdog resets of the system etc. It sends set-points for thruster angular velocities to the MCU board and receives back the actual thruster angular velocities. Although the control loop for the thruster angular velocities is closed on the MCU board, the Main Computer needs the readings due the latent response, so as to use it as its actual control signal for a more accurate observer. The Main Computer runs a real-time-modified Linux system and the MCU runs a single process loaded in its flash memory.

VI. Hovercraft Battery Management System

A. Overview

The hovercraft battery management system allows HoTDeC to be completely self-sufficient. This system enables the hovercraft to detect when their battery systems require recharging and when necessary automatically connect to an external power supply for charging without any human contact or external signaling.

B. Operation

When the batteries require charging the SBC receives a signal from the battery manager signaling to return to its docking station. The docking station is contains two electric contacts that provide both charging and operation power to the hovercraft. These electric contacts mate with two contacts on the bottom of the hovercraft body. The skirt design conveniently contains venting holds in the bottom of it allowing two electrical probes to make contact with the docking station only when the skirt is deflated and the hovercraft is parked in the docking station. When the hovercraft requires a recharge or when not in use the

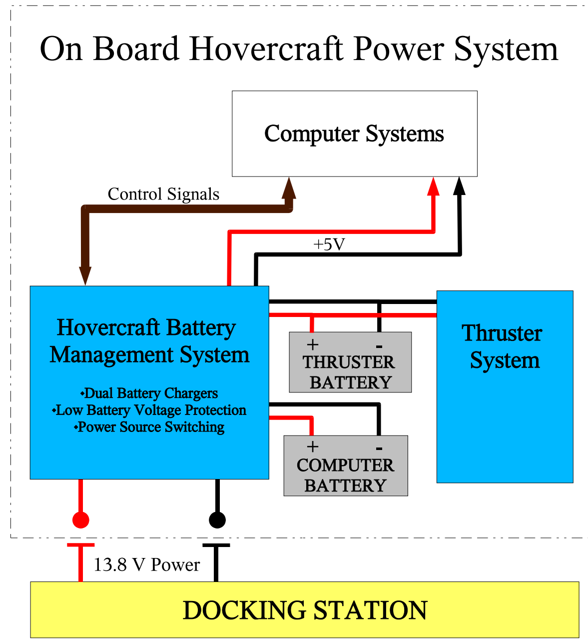

Fig. 5. Hovercraft Battery Management System Block Diagram

hovercraft controller simply moves the vehicle over to a docking station and parks itself over the electrical contacts, when the skirt is deflated the entire weight of the hovercraft is supported by these two electrodes insuring a secure electrical connection.

When the battery management system detects that the docking station power is present and available, a signal is sent to the SBC. Once this signal is received by the SBC, the SBC will disable the thrusters and signal the battery manager to transfer over to docking station power and commence battery charging. The battery manger is able to provide seamless hot swapping of power sources allowing the SBC and other communications electronics to operate without interruption.

Two separate chargers then charge the thruster and SBC battery systems according to preprogrammed charge current limits appropriate to th e batteries installed. The chargers are designed to provide the ideal charging scheme for lithium polymer batteries, a constant current is used to charge until the battery has reached the max charge voltage, this is then followed by constant voltage at that set point. During charging the battery manager reports both the batteries charge voltage and the charging current to the SBC. This information allows for very accurate assessments of the batteries capacity and time remaining for a complete charge. The system will automatically terminate charging when the charging current drops below a programmed threshold. With the current battery configuration the hovercraft can operate for 15 minutes at a time before exhausting its batteries. A completely exhausted hovercraft can be recharged in as little as 150 minutes. When the unit is docked and the SBC has authorized the use of docking power, there is no draw on the batteries all power is provided by the docking station. As an additional safety if external power is ever lost, the hovercraft will automatically default to battery power regardless of the SBC settings.

There is no need to wait for the hovercraft to completely recharge for before running additional experiments. Anytime during the charging period a user can override the automatic charging sequence and run a control on a hovercraft with whatever power is available in its battery system. The battery manger continuously reports the voltage and charge current to the SBC, which then in turn transmits it to the web server. Using calculations of the vehicle's current consumption during a particular experiment and current consumption rate combined with the voltage on the batteries, a very accurate calculation of the power remaining for a particular hovercraft mission is possible. The software automatically verifies that the projected power in a hovercraft's battery packs is sufficient to complete the entire duration of the controller experiment that the user has entered.

VII. MECHANICAL VEHICLE SPECIFICATION AND FUNCTIONALITY

To accommodate the SBC and the battery management system the HoTDeC hovercraft vehicle chassis was redesigned using CAD modeling software. The new construction allows for much faster manufacturing and assembly of the hovercraft since little or no hand processing is required to assemble the chassis pieces that are produced using a CNC controlled three axis mill at UIUC. The final piece is a very light aesthetic cover requiring minimal fastening that finishes the craft hiding the electronics and holds the on-board camera or hovercraft-identifying top pattern used in ceiling mounted camera experiments.

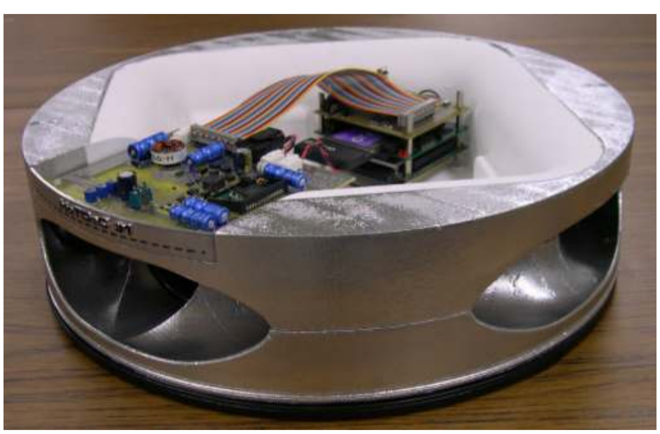

Fig. 6. A HoTDeC vehicle - top removed

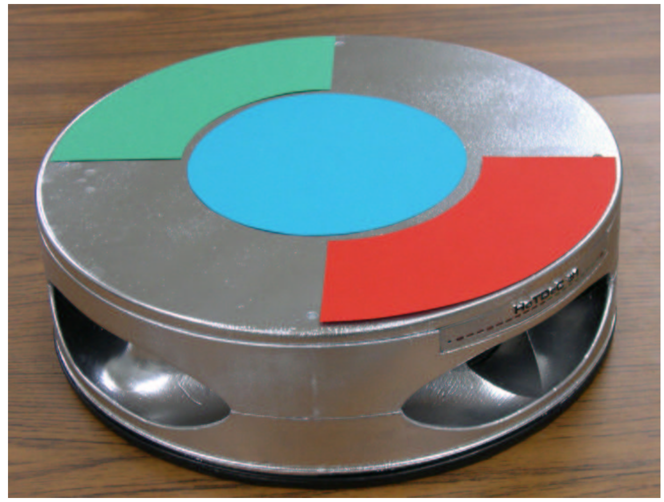

Fig. 7. A HoTDeC vehicle, with the vision indicator top

VIII. FUTURE WORK

A. Modularity of the System

Several parts of our testbed are entirely modular, and can be replaced or interchanged with other systems' modules.

1. The hovercraft's computer system can be used on other mechatronics projects as-is, or be replaced with a similar one.

2. The charging stations, batteries and power boards can be used on standard systems that run 5V/13V devices, or be redesigned without the function of the rest of the system being affected.

3. The vision system can give feedback regarding the location of the HoTDeC color-coded top boards, which can be mounted on any vehicle that can handle their size (currently 16-inch in diameter). Any vision system that can handle the number of hovercraft in the testbed and the 30Hz requirement can be swapped in.

B. Expansions

The two forthcoming expansions are regarding on-board sensors. We are planning for inertial sensors in the short term, and on-board vision after that.

1) Inertial Sensors: The estimation of the hovercraft states is currently done using the vision and thruster speed data. To increase the accuracy of this estimate an inertial navigation unit is being built. This unit will consist of three accelerometers and three angular velocity sensors as shown in figure 8.

2) On-board Vision: The next stage in the development of the HoTDeC vision system includes on-board vision. This feature removes the need for external ceiling mounted cameras and allows the hovercraft to travel throughout corridors and large rooms. The light-weight IEEE 1394 camera is well suited to the task of providing on-board vision given that the single board computer running Linux on the hovercraft will be able to operate code similar to that used for the overhead cameras. Finally, using the flexible, on-board IEEE 1394 camera system in conjunction with a hyperbolic mirror, 360 degree vision can be implemented to allow the hovercraft to see and move autonomously in all directions.

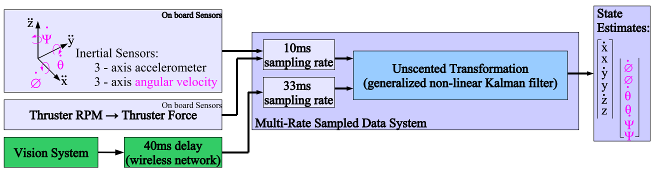

Fig. 8. Estimation with Inertial Sensors

The estimator uses a method of removing the correlation resulting from the integration of the accelerometer signals from Gelb [1]. This method is combined with Julier's generalization of the Kalman filter [2] and used to estimate the states of the non-linear system without linearizing the system.

Current work consists of the combination of this concept with the concept of the multi-rate problem in Lall [3].

REFERENCES

[1] Arthur Gelb, *Applied optimal estimation*, The Analytic Sciences Corporation, 1986.

[2] Simon Julier, Jeffrey Uhlmann, and Hugh F. Durrant-Whyte, *A new method for the nonlinear transformation of means and covariance in filters and estimators*, IEEE Transactions on Automatic Control **45** (2000), no. 3.

[3] Sanjay Lall, and Geir Dullerud, *An LMI solution to the robust synthesis problem for multi-rate sampled-data systems*, Automatica **37** (2001).

[4] Stubbs and Dullerud: Networked control of Distributed Systems: a Testbed *ASME 2001 International Mechanical Engineering Congress and Exposition*

[5] Stubbs, Vladimerou, Vaughn and Dullerud: Design of a Vehicle Network Control Testbed *ACC 2002*

[6] Stubbs, Vladimerou, Rubel and Dullerud: Distributed Control of Network Vehicles *CDC 2002*

[7] *Bluetooth* Specification Version 1.1, Bluetooth SIG, 2001.

[8] *IEE 802.11* specification, the IEE 802.11 WIG

[9] Caltech's *MVWT*, *MVWT-II*
`http://www.cds.caltech.edu/~mvwt/`

[10] Formation Flying at MIT
`http://www.mit.edu/people/jhow/ff.html`

[11] *Roboflag*
`http://roboflag.mae.cornell.edu/`

[12] *Camera Calibration Toolbox for Matlab*, Jean-Yves Bouguet, MRL - Intel Corp.
`http://www.vision.caltech.edu/bouguetj/calib_doc/`