CACSD Practical Session
Rotary Inverted Pendulum

Oscar Barrero – Jeroen Buijs – Bart De Moor
ESAT-SISTA (room 02.12)
Kasteelpark Arenberg 10, 3001 Heverlee
e-mail: oscar.barrero@esat.kuleuven.ac.be
www.esat.kuleuven.ac.be/education/options/da/iroptiondna/vakhomepage.php?id=5
tel: (016)32 11 60
This manual was written by Peter De Gersem and Jeroen Buijs

2002-2003

1 Aim of this Session

The subject of this session is the *Rotary Inverted Pendulum*. This setup consists of a rod (the pendulum), mounted on an arm which can rotate in a horizontal plane, driven by a motor (fig. 1). As you can readily imagine, this system has an unstable equilibrium point (the rod staying straight up). You'll have to design a stabilizing controller for this Rotary Inverted Pendulum, i.e. a controller which drives the motor such that the rod stays vertical and the arm goes to, and stays at, a desired angle. The measured signals are the angle of the arm and the angle of the rod (both obtained with potentiometers), the output is the voltage to the motor. The controller is implemented in software on a PC, equipped with a data-acquisition card.

It's important that you read this manual completely before starting. And read section 3 again before doing experiments on the real setup.

2 Control Strategy

First of all, the CACSD course is concerned with *model-based* control system design. So the first thing to do is to find the *state-space* model of the system. Next, you can apply the model-based methods described in the course notes to design the controller. You will need a computer program to do so (MATLAB), and in the meantime you can do some simulations and calculations to predict the response of the *closed-loop* system[1] using SIMULINK. When you're convinced that your controller has the right properties, it's time to try it out on the real system.

[1] The closed-loop system is the system consisting of the plant and the controller.

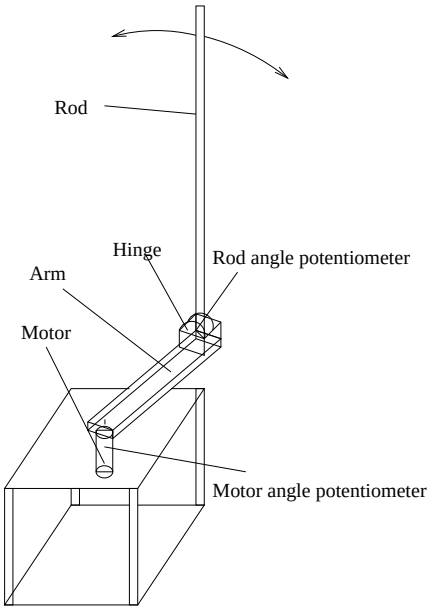

Figure 1: The rotary inverted pendulum setup.

2.1 The State-space Model

There are different possibilities to obtain the state-space model of a physical system. One of them is identification. In this case however, physical modeling, i.e. using some mechanical and electrical laws you've probably already forgotten, is simple enough, so that's the way we will do it. The derivation of the model is given, but make sure that you understand what is happening, not only because we might ask you some questions about it, but especially since some insight in how the model is derived may lead to a better understanding of the results you obtain later on. It's logical that good knowledge of the system you're going to control and of the model you use to represent this system, is the first step towards a well functioning controller.

First, we will model the mechanical part of the system. In a second step the electrical part will be modeled and the two models will be concatenated.

2.1.1 The Mechanical Part

You can simplify the mechanical modeling by neglecting friction, and by considering the mass of the rod to be concentrated in the middle of the rod. Fig. 2 shows a drawing of this simplified model. The input of the mechanical system is the torque T (coming from the motor), the outputs are the angle of the arm θ and the angle of the rod α. As you might discover, the state-space model of this system contains four states. The most logical choice seems $\{\theta, \alpha, \dot{\theta}, \dot{\alpha}\}$, so we will stick to these.

We will use the Lagrange equations here to derive the non-linear model.

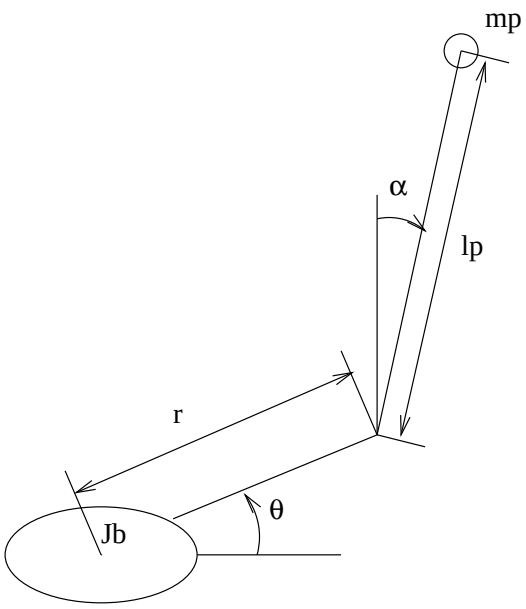

Figure 2: The simplified model of the inverted pendulum setup (with l_p being half of the real length of the rod). Watch out: unless how it is drawn, the two angles are positive in the SAME direction, meaning that in the equation of the absolute pendulum speed, all signs will be positive.

They are given by

$$\begin{aligned} \frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{\alpha}}\right) - \frac{\partial L}{\partial \alpha} &= 0 \\ \frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} &= T \end{aligned}$$

where the Lagrangian $L = E_{kin,tot} - E_{pot,tot}$ and T is the torque coming from the motor. The kinetic energy is the sum of the kinetic energy contained in the arm and the kinetic energy contained in the rod. The total potential energy can easily be formulated as

$$E_{pot,tot} = \frac{mgl\cos\alpha}{2}.$$

Notice that the rod is represented by a mass at half its length, so l is half the rod length (the value given at the end of this manual also represents half the rod length!).

The kinetic energy contained in the arm is

$$E_{kin,arm} = \frac{J_b\dot{\theta}^2}{2}$$

with J_b the inertia of the arm (at the rotating axis). The kinetic energy of the rod is given by

$$E_{kin,rod} = \frac{mv_a^2}{2}$$

where m is the rod's mass and v_a is the absolute velocity of the rod's center. This velocity can be calculated in different ways. Here we prefer to calculate

the position of the rod's center and take its derivative towards time to calculate v_a. Therefore we introduce a coördinate system (X, Y, Z) with the (X, Y)-plane in the horizontal plane and the origin at the rotation center of the arm. Let the X-axis being the axis where $\theta = 0$ and let the Z-axis point upwards. Then the position of the rod's center is given by

$$\begin{aligned} X_r &= r\cos\theta - l\sin\alpha\sin\theta \\ Y_r &= r\sin\theta + l\sin\alpha\cos\theta \\ Z_r &= l\cos\alpha \end{aligned}$$

and after some basic calculations, the square of the absolute velocity can be found to be

$$v_a^2 = r^2\dot{\theta}^2 + l^2\dot{\alpha}^2 + l^2\dot{\theta}^2\sin^2\alpha + 2rl\dot{\theta}\dot{\alpha}\cos\alpha$$

which leads to the following expression of L,

$$L = \frac{J_b\dot{\theta}^2}{2} + \frac{m(r^2\dot{\theta}^2 + l^2\dot{\alpha}^2 + l^2\dot{\theta}^2\sin^2\alpha + 2rl\dot{\theta}\dot{\alpha}\cos\alpha)^2}{2} - \frac{mgl\cos\alpha}{2}.$$

The Lagrange equations given above now lead to the following non-linear model equations,

$$\begin{aligned} (mrl\cos\alpha)\ddot{\theta} + ml^2\ddot{\alpha} - ml^2\dot{\theta}^2\sin\alpha\cos\alpha - mgl\sin\alpha &= 0 \quad (1) \\ (J_b + ml^2\sin^2\alpha + mr^2)\ddot{\theta} + (mrl\cos\alpha)\ddot{\alpha} + ml^2\dot{\theta}\dot{\alpha}\sin 2\alpha - mrl\dot{\alpha}^2\sin\alpha &= T \quad (2) \end{aligned}$$

2.1.2 The Electrical Part

Since our control input is not the torque T, but the electrical voltage V applied to the motor that generates this torque, we have to eliminate T by introducing the electrical equations of the motor. The equations governing a DC-motor are

$$V = IR_m + K_b\omega_m$$

$$T_m = K_m I\,,$$

in which R_m is the *Armature resistance*, K_b the *Back EMF constant*, K_m is the *Motor torque constant*, T_m is the torque produced by the motor at its shaft. The meaning of V, I and ω_m is clear, I hope. The motor drives a gearbox of ratio K_g (i.e. $\omega_m = K_g\omega_g = K_g\dot{\theta}$ and $T_g = K_g T_m$, in which ω_g and T_g are the rotation speed and the torque at the output of the gear). The torque T_g is of course the torque T in equation (2). Concluding, the relation between T and the applied voltage V is given by

$$T = \frac{K_m K_g}{R_m}V - \frac{K_m K_b K_g^2}{R_m}\dot{\theta} \quad (3)$$

Eq. (3) can be used to eliminate T from (2).

2.1.3 The Global Model

Solving equations (1) and (2) for $\ddot{\alpha}$ and $\ddot{\theta}$, we obtain a non-linear state space model. Since the methods we will use to control this system are based on linear models, we must linearize the obtained model around the non-stable equilibrium point. Doing so (think a second about how this is done), we obtain a standard linear state equation of the form $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$,

$$\begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \\ \ddot{\theta} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-mrg}{J_b} & \frac{-K_g^2 K_m K_b}{R_m J_b} & 0 \\ 0 & \frac{(J_b+mr^2)g}{J_b l} & \frac{K_g^2 K_m K_b r}{J_b R_m l} & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \alpha \\ \dot{\theta} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{K_m K_g}{J_b R_m} \\ \frac{-K_m K_g r}{J_b R_m l} \end{bmatrix} V$$

Adding the output equation, you obtain a global state space model. The values of the different constants for the rotary inverted pendulum are given at the end of this manual. Make sure to convert them to the right units. Now the model is known, make a short open loop analysis of the system and the model, and try to formulate the control problem.

2.2 The Controller

The controller you're going to design is of LQR-type. The controller will be **designed in continuous time** and afterwards converted to discrete time, to be implemented on a digital computer. Because the states θ and α are measured, and $\dot{\theta}$ and $\dot{\alpha}$ can be easily calculated from these measurements, no state-observer is needed.

The LQR-controller determines an optimal state-feedback gain K, such that the closed-loop system $A - BK$ (i.e. $\mathbf{u} = -K\mathbf{x}$) minimizes the quadratic performance index

$$J = \int (\mathbf{x}^t Q \mathbf{x} + \mathbf{u}^t R \mathbf{u}) dt \,.$$

The weight-matrices Q and R determine the relative importance of minimizing the states and the input-signal. You can take

$$Q = \begin{pmatrix} 0.25 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

and

$$R = 0.05$$

as initial values[2] (this assumes your states are $\mathbf{x} = \begin{bmatrix} \theta & \alpha & \dot{\theta} & \dot{\alpha} \end{bmatrix}^t$, and that your state-space matrices are expressed in SI units). Use the MATLAB-command[3] `lqr` to calculate K.

[2] Also calculate a controller with $Q = \mathrm{diag}(0.25, 4, 0, 1)$, this seems to give better results on the actual setup. Is the original controller stable or unstable, in theory and in practice?

[3] To figure out how a MATLAB-command works, type `help` *command* at the MATLAB-prompt. Also if you don't know how to do something in MATLAB, you should try `help`. If you're really stuck, you can always mail me for suggestions.

It might not be clear to you how to apply an input to the closed loop system, and how to simulate that in SIMULINK (e.g. for the step- or impulse-response): to position the arm at a certain angle, we have to tell the LQR-controller that the θ-state has to go to θ_{desired} instead of zero. To obtain that behavior, we calculate the state-feedback not with $\mathbf{u} = -K\mathbf{x}$, but with $\mathbf{u} = -K(\mathbf{x} - \mathbf{x_d})$, where $\mathbf{x_d}$ equals $\begin{bmatrix} \theta_{\text{desired}} & 0 & 0 & 0 \end{bmatrix}$. So the input to your controller block K in SIMULINK has to be the difference between x and x_d[4].

While designing a controller, you need to evaluate the performance of your design, to be able to compare it with other designs. Some *evaluation criteria* are

- the **closed-loop eigenvalues (or pole-zero plot)**. They give information about the stability-margin, the damping and the eigenfrequency of the closed-loop system. The closed-loop system should at least be stable.
- the **step-response**. This gives information about settling-time, rise-time, overshoot, amount of input-signal...
- the **impulse-response**. This gives information about the damping, the eigenfrequencies of the closed-loop system,...

With LQR, the choice of the weight-matrices is the most difficult part of the design. The strategy proposed here is simply *trial-and-error*: start with the initial values, evaluate the prestations, change one parameter and evaluate, change another parameter,... The ideal controller is stable, robust (can recover from a large disturbance), and is not (too) nervous or noise-sensitive.

It's up to you to find good weight-matrices with trial-and-error. Keep a record of your different trials (make plots of the evaluation criteria - see section 'Report'), and argument why you are satisfied with your final design. Put the representative information in your report (see section 4). Don't spend *too* much time on this: **it's important that you have a satisfying controller, and that you understand what the effect is of changing parameters in Q and R.** It's not the aim of this practical session to try as much different values for Q and R as possible.

The result of your iterative design is a state-feedback gain K. The controller-program needs the feedback-gains in V/deg, and not in V/rad. Be sure you make the right conversions! You can check the response in your simulations.

2.3 Implementation Details

The controller has to be implemented on a digital computer, in software. Therefore the inputs are measured by means of a biased potentiometer[5] and a data-acquisition card in the computer: the voltage at the wiper-pin of the potentiometer is a measure for the angle of the arm and the angle of the rod. This

[4] Another approach would be to calculate one closed-loop state space system from (A, B, C, D) and K with input x_d. But nevermind.

[5] The two ends of the potentiometer are connected to the $+12$V and -12V terminals of the Power Supply, and the wiper-pin of the potentiometer is connected to the arm resp. the rod.

voltages are converted to a digital value by an A/D-converter. This digital values are read by the controller (which is a computer program, written in ordinary C), and are converted from voltage to centimeters resp. degrees (the relationship is linear: -5V to $+5$V corresponds with -176 to $+176$ degrees). The controller samples these voltages at a frequency of 200 Hz (that's an arbitrary choice: fast enough compared to the dynamics of the system, not too fast for the computer to handle). The states $\dot{\theta}$ and $\dot{\alpha}$ are calculated using the backward difference equation:

$$\dot{y} = \frac{y^f(t) - y^f(t - T_s)}{T_s},$$

where T_s is 1/200 s. To use this technique, it's necessary to low-pass filter the raw measurements before calculating the difference, because these are very fluctuation- and noise-sensitive (that's why there's y^f instead of y). In the controller-program, you can try different values for the cut-off frequency ω_c of this low-pass filter to study the effects. A typical value is 2 Hz ($\omega_c = 2 \cdot 2\pi$ rad/s). The low-pass filter obeys the following law:

$$y_n^f = \frac{\omega_c T_s}{1 + \omega_c T_s} y_n + \frac{1}{1 + \omega_c T_s} y_{n-1}^f .$$

Make sure you understand what this filter is doing so that you understand where exactly the outputs should be filtered, since you will have to include this filter in your simulation schemes. Hint: Typically 50% of the students is doing this wrong.

Then the output voltage is calculated as $V = -K\mathbf{x}$ or $V = -K_1(\theta - \theta_{\text{desired}}) - K_2\alpha - K_3\dot{x} - K_4\dot{\alpha}$. Of course, the output voltage is limited: the controller limits it to ± 5V. This voltage is then converted to an analog value with the D/A-converter, and a power-amplifier applies this voltage to the motor of the arm.

As stated before, the controller itself is a C-program. The main function is only the user interface, the controller is implemented in an interrupt-service routine that is fired each T_s seconds by programming the clock-chip of the PC. If you're interested, the source-code is on the computer too.

3 The Proof of the Pudding

Now it's time to implement your controller. The controller-program has been written for you, the only thing you have to do is to enter your state-feedback gains and to see what happens. You should plan your experiments before you come to the setup, because the time is limited. Try to spend your time at the setup as efficient as possible.

The actual setup resides in room 02.58. On a table you find the frame with the arm and the rod, a power module, and a PC. Before you do anything, read these instructions thoroughly, because the setup is not that robust. **To use the set-up, you'll have to make a reservation first**. This can be done

automatically on the net[6].. The set-up will also be available during 'Christmas holidays weeks'. Normally the best thing to do is to make a reservation for two up to four hours, just in case something goes wrong... More info about what to do once you've entered the ESAT-building when you want to do some tests can be found on the net.

If something happens, or something doesn't seem normal, don't hesitate to call me (internal number 1160). Also remember to regularly check the home-page for the CACSD-course (see title-page) for commonly asked questions and other problems.

3.1 Description of the software

The controller-program provides a user interface that lets you turn on/off the motor, fill in the gains, and plot and save the measurements. When you start up the program, the motor is off and the default values are displayed. In this screen, you have the following possibilities:

'o' Toggle the motor on and off,

'k' Enter the gains,

'p' Set the desired angle of the arm (in deg, relative to the current zero),

'e' Toggle the desired angle to its negative,

'z' Take the current measurement as zero (necessary to bias the potentiometers correctly); to use this, turn the motor off, turn the arm back to zero degrees, hold the rod in the equilibrium position, press 'z', set the desired position to 0 ('p'), and turn the motor back on,

'r' Go to realtime plot,

'q' Quit the program, the motor will be turned off,

'f' Enter the cutoff frequency of the low pass filter (for calculating the differential); this command can be used to study what the controller would do without filter (take a large cutoff frequency, e.g. 1000 Hz).

In the realtime plot-screen you have the same commands (except for 'z', 'k' and 'r'), and you have some more:

't' Enter the length (in seconds) of one screen,

'v' Plot the next variable (Arm angle, Arm angular velocity, Output volts, Rod angle, Rod angular velocity),

'x' Return to the text screen,

[6] `http://www.esat.kuleuven.ac.be/~belleman/cgi-bin/table.pl`

's' Save the current screen to a MATLAB m-file; when you press the key, nothing happens until the screen is full; then the program asks you if you really want to save the data, it warns you that the motor will be turned off and that you have to grasp the rod, it asks you for the filename (preferably on your own diskette, with extension .m because it's a MATLAB m-file), saves *all* the variables, and returns to the realtime screen; you have to turn the motor back on yourself, but keep the rod in equilibrium position while doing this.

Please save the data on your own diskette, because otherwise the `c:\cacsd` directory will become a mess. There's no guarantee that data on the hard disk will remain there...

3.2 Try it out

When you're ready for it, follow this guideline:

1. If the setup is not ready, or if you think there's something wrong, call me (1160 internal). It's a good idea to check for loose cables. Put the rod on the arm, and fix the screw as tight as possible (by hand).

2. Turn the computer on, change directory to `c:\cacsd`.

3. Turn the Power Module PA0103 on (by pressing the red ON/OFF button).

4. Start the controller by typing `rotpend`. You should see the opening screen by now.

5. Turn the arm to zero degrees, hold the rod as vertical as possible (try to put it in its equilibrium point) and press 'z' to give the zero-reference. Check the values in the display while moving the arm and the rod slightly. The values should be realistic.

6. Enter your gain values (by pressing 'kn' where n is 1, 2, 3 or 4, and typing the right value). Be careful that you choose the right units (using cm and degrees).

7. Hold the arm and the rod around equilibrium position (the display should indicate almost zero for both measurements), and turn on the motor (by pressing 'o'). **Always hold the frame**, because it can tilt if moving too fast or wildly. If the system goes unstable, turn the motor back off (quick), grasp the rod before it falls, and reconsider your design. Maybe you used the wrong signs for the gains. Also think about the magnitude of the gains, e.g. what voltage is generated if the position is 1 mm of equilibrium...

   If the system is oscillating a little, don't care about it. However if it's completely stable, you deserve some congratulations!

   If the rods stays straight up, try disturbing it slightly (hold the frame!) and see what happens. If that works too, try changing the angle (by pressing 'p', start with 25 deg or so) and see what happens (hold the

frame!). If everything goes well, take a look at the realtime plot and save some representative plots. You can use 'e' to toggle the set point easily.

If the system is in equilibrium, it may happen that the angles are not exactly zero. Can you figure out what's wrong, and what you can do about it?

You should vary the feedback-gains a little to compare the response. Change one at the time and explain what happens. Be careful, because the system can go unstable!

8. If you've finished, quit the controller, turn the power module off, and turn the computer off.

4 Report

It might be a good idea to read this section before you start your calculations, simulations and tests. It might give you an idea of the things you should do and the things you should't waste time on.

So at the end, the bad news: you will have to write a report about your design and the results. I prefer a printed report, which may be in English or Dutch (French is no option...). One report per group suffices. This report will be used at the oral "examination" (see further).

Contents of the report:

Section 1. The model
As an introduction, give the linear state space model and do a short open loop analysis of this model. This should lead to a formulation of the control problem.

Section 2. Building the LQR controller
Build two different simulation schemes in SIMULINK, which **I want you to send me by mail when you enter your report**. The first just has to contain the linear model (continuous time) and the controller. In the second, you must include as much as possible: A/D convertors, the cut-off filter to calculate the derivatives (notice that the second simulation scheme will have a different C matrix in its model!!!), the saturation, noise, ... (don't try to include friction).

First build the controller with the Q and R as given before in this manual. Analyse this controller with the two schemes, using the evaluation criteria as mentioned before (make a distinction between frequency domain and time domain analysis). Then try to change the different values of Q and R and look at the result. Give an explanation about the different advantages and disadvantages of changing these values.

Also shortly discuss the influence, (dis)advantages, etc. of the different blocks in the second scheme (saturation, cut-off filter, etc.).

Out of these tests (don't exaggerate in their amount: once you can tell what the influence is of the different values in the tuning parameters and you have found a good enough controller, you can stop), choose one controller to implement and tell why you've chosen this particular one.

Section 3. The proof of the pudding
Well, let's see how well your controller really is now. Use the chosen K-values to use on the test setup in room 02.58. Make some representative plots of the inputs and the states for a step in θ and preferrably also for a small(!) kick against the rod (CAUTION: HOLD THE FRAME!) . Try to explain the main differences between simulation and real test. If you think it's necessary (and be sure that for the rotary inverted pendulum the probability is pretty high that it will be), try to change your K-values a little bit to obtain a better controller. Explain what the result is of a certain change in K.

Also try to explain the role of the cut-off frequency (what if you make it higher, resp. lower) and what happens when your calibration isn't perfect. If you notice other at first sight strange things, try to think about them and mention them in your report. This may become handy during examination.

Section 4. Conclusions
Every well written report ends with this section, so we won't make an exception here.

5 Examination

Once the report is finished, you can give me a copy. The deadline is two days before the oral examination. Don't forget to send me the Simulink-schemes together with the necessary m-files as well. The examination will include some questions about the report (things that are not mentioned or unclear, but also things that are mentioned, just to be sure you didn't just copy your brother's results and report of some years ago), but it's good to know that it is evaluated separately from the report (although the remark should be made that without some kind of report, there will be less opportunity for scoring on the oral examination by an obvious lack of questions).

Further information (dates of examination, how the points are distributed between exercises and practicum, resp. report and oral examination, etc.) can be found using 'the fastest way to get from where you wouldn't want to be to where you wouldn't want to go'[7].

6 Specifications

This section contains some data on the rotary inverted pendulum setup.

Motor armature resistance : $R_m = 2.6\,\Omega$

[7] i.e. Internet (Andrew Eldritch, Sisters of Mercy)

Motor torque constant : $K_m = 0.00767 \ \frac{\mathrm{N\,m}}{\mathrm{A}}$

Motor back EMF constant : $K_b = 0.00767 \ \frac{\mathrm{V}}{\mathrm{rad\,s}}$

Motor gear ratio : $K_g = 70 : 1$ (The output is slower)

Arm inertia : $J_b = 0.0044 \ \mathrm{kg\,m^2}$ (at the rotating axis!)

Arm length : $r = 0.145 \ \mathrm{m}$

Rod length : $l = \frac{l_{tot}}{2} = 30.5 \ \mathrm{cm}$

Rod mass : $m = 0.210 \ \mathrm{kg}$